PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-149599
(43)Date of publication of application : 27.05.1994

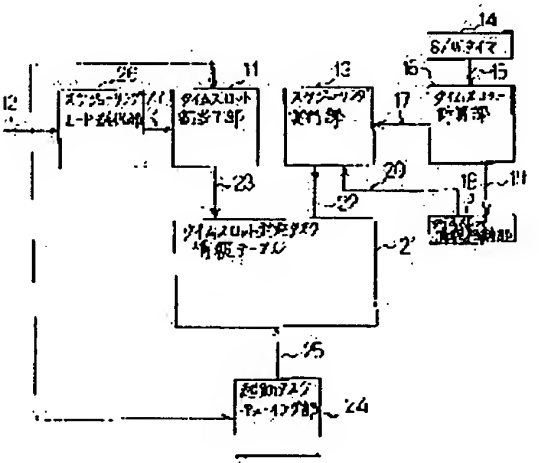
(51)Int.Cl. G06F 9/46

(21)Application number : 04-327449
(22)Date of filing : 12.11.1992

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : NAKAMURA MASAHIRO

(54) COMPUTER SYSTEM

(57)Abstract:
PURPOSE: To provide the computer system which can schedule tasks of all periods by providing scheduling in which case conventional periodical scheduling cannot be scheduled and there are tasks having execution periods shorter than a minimum reference period.
CONSTITUTION: This computer system consists of a scheduling mode selecting part 26 and a time slot assigning part 11 in addition to the system constituting the conventional scheduling system. The scheduling mode selecting part 26 changes the scheduling mode from the periodical scheduling mode to the priority when there are tasks existed that are execution periods shorter than the minimum reference period which becomes the reference period at the time of periodically checking the activation of tasks by a scheduling execution part 13 in the activation requesting tasks. The time slot assigning part 11 generates table information 23 of scheduling in accordance with the selected scheduling mode.



LEGAL STATUS

[Date of request for examination]	28.09.1995
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	2760721
[Date of registration]	20.03.1998
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

Japanese Unexamined Patent Application Publication No. 6-149599

SPECIFICATION <EXCERPT>

[0018] FIG. 2 is a functional block diagram showing an internal structure of the above-mentioned scheduling mode selecting unit 26. In FIG. 2, a reference numeral 26a is an execution period determining unit which determines an execution period of each task based on task information 12. A reference numeral 26b is a classifying unit which classifies each task into one of execution levels ranging from a level with a short execution period to a level with a long execution period based on the result of the determination by the execution period determining unit 26a. A reference numeral 26c is a comparing unit which judges whether or not an execution period of an execution level, into which a task with a minimum execution period is classified by the classifying unit 26b, is shorter than the above-mentioned minimum reference period. A reference numeral 26d is a scheduling mode switching unit which switches between a conventional periodical scheduling mode and a priority scheduling mode, which is a new scheduling mode, based on the result of the comparison. The priority scheduling mode assigns an execution level, into which a task with a minimum execution period is classified, to every time slot. At that time, an output from the scheduling mode switching unit 26d is a scheduling mode signal 27. In addition, the execution period determining unit 26a and the classifying unit 26b correspond to an execution period determining unit 11a and a classifying unit 11b, respectively, which are originally included in a time slot assigning unit 11 (see FIG.17). Therefore, it is also possible to have only one set of these so as to be shared by

the time slot assigning unit 11 and the scheduling mode selecting unit 26.

[0019] Next, the time slot assigning method according to the present embodiment is explained with flow charts FIG.3 and FIG.4. First, the scheduling mode selecting unit 26 determines execution periods of all active tasks by using the execution period determining unit 26a, and classifies the active tasks according to their execution periods into the execution levels by using the classifying unit 26b (ST31). For example, a task with an execution period 50 ms is classified into an execution level 1, and a task with execution period 70 ms is classified into an execution level 2. Next, the scheduling mode selecting unit 26 compares the minimum reference period and the execution period in the execution level 1, to which the task with the minimum execution period belongs, by using the comparing unit 26c (ST32). If the execution period in the execution level 1 is smaller, the priority scheduling mode is selected by the scheduling mode switching unit 26d for switching (ST34). On the other hand, if it is larger, the conventional scheduling mode is selected for switching (ST33). Next, the time slot assigning unit 11 receives the scheduling mode signal 27 and checks its mode (ST35). In the case of the priority scheduling mode, as shown in FIG. 5, the time slot assigning unit 11 assigns the execution level 1 to all of the time slots 1 to N in a time slot managing table 211 (ST36). In the case of the periodical scheduling mode, a processing is performed in the same way as the conventional system. That is, the time slot assigning unit 11 performs the following processing with respect to execution levels 1 to M (ST37). As for the execution level 1, multiplying the sum of all of the task execution time in the execution level 1 by the ratio of the execution period in the execution level 1 to the minimum reference period gives the time which the total execution time of the tasks in the execution level 1 takes in the minimum reference period (ST38). As described in the

conventional example, for example, in the case where an execution period of tasks in the execution level 1 is 100 ms, the total execution time in the execution level 1 is 40 ms, and the minimum reference period is 50 ms, 40 x (50/100) = 20 ms is the obtained value . That is, the execution level 1 requires an execution time for 20 ms within the minimum reference period. By dividing the obtained value by a time-slot time and rounding up, the number of time slots required in the execution level 1 within the minimum reference period is obtained (ST39). In the above example, when time-slot time is 10 ms, 20 / 10 = 2, which means two time slots are needed. The number of time slots required in each execution level is obtained by repeating the processing as described above from execution levels 2 to M (ST40). Next, the number of time slots required for the execution time of the system which is needed for executing a task (ST41). The number of time slots required in each execution level is calculated with the above procedure. However, when the sum of the number of time slots exceeds the number N of the time slots in the minimum reference period, the maximum execution load is exceeded, which results in losing control. In order to avoid this, the sum is compared to N (ST42), then if the sum is over the number of time slots N, time slots just as many as needed is stored in the time slot managing table 211 sequentially from the execution level 1. Then, at the time when the table is filled, the remaining levels are ignored (ST44). When N is larger than the sum, the time slots as much as needed is stored in the time slot managing table 211 sequentially from the execution level 1 without any problems(ST43). The assignment of the execution levels to the time slots is completed as above.

[0020] A scheduling execution after the time slot assignment is performed in the same operation as the conventional way. In this regard, in the case where the time slot managing table 211 is generated in the priority scheduling mode as shown in FIG. 5, the

flow chart of FIG. 21 mentioned in the conventional example is changed to FIG. 6. Every time slot acquires execution level 1 (ST212), level managing tables 212a-212c are checked from the execution level 1 as shown in FIG. 22, and a queuing task found at first is executed. As a result, the maximum priority is given to the execution level 1 and the minimum priority is given to the execution level M. That is, because every time slot acquires the execution level 1, and an task to be executed is determined according to the result of a search for the execution level as shown in FIG. 22, a task in a higher execution level is not executed as long as a task in a lower execution level is not executed. Consequently, the priority scheduling, in which a task in smaller execution level is given a higher priority, is provided. In this case, FIG. 16 mentioned in the conventional example is changed to FIG. 7.

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－149599

(43)公開日　平成６年(1994)５月27日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/46	３４０　Ｂ	8120－5B		

審査請求　未請求　請求項の数３（全 30 頁）

(21)出願番号　特願平4－327449

(22)出願日　平成４年(1992)11月12日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　中村　正博
神戸市兵庫区和田崎町１丁目１番２号　三菱電機株式会社制御製作所内

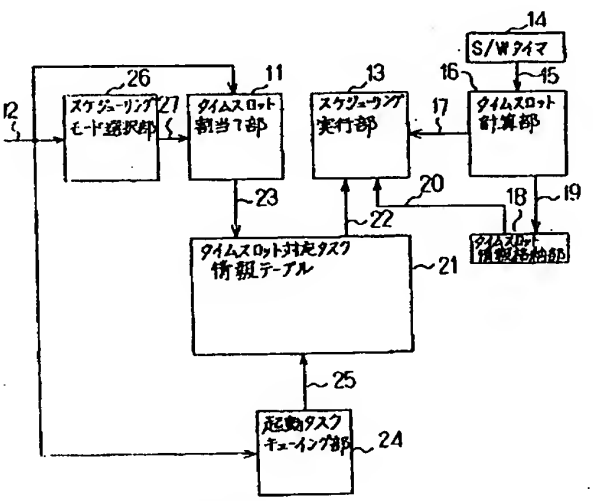
(74)代理人　弁理士　宮園　純一

(54)【発明の名称】　計算機システム

(57)【要約】

【目的】　従来の周期的スケジューリングでは不可能な、実行周期が最小基準周期未満のタスクが存在する場合のスケジューリングを提供し、全ての周期のタスクのスケジューリングを可能にした計算機システムを得ることを目的とする。

【構成】　従来のスケジューリング方式を構成するシステムに加えて、起動要求タスク中に、スケジューリング実行部１３が周期的にタスクの起動をチェックする際の基準周期となる最小基準周期未満の実行周期のものが存在する場合、スケジューリングモードを周期的スケジューリングモードからプライオリティスケジューリングモードに変更するスケジューリングモード選択部２６と、選択されたスケジューリングモードに従いスケジューリングのテーブル情報２３を生成するタイムスロット割当て部１１から構成される。



12；タスク情報
15；定周期割込み
17；タイムスロット変更割込み
19；タイムスロット値(変更値)
20；タイムスロット値(現在値)
22；テーブル情報(現在値)
23；テーブル情報(変更値)
25；起動要求タスク
27；スケジューリングモード信号

1

【特許請求の範囲】

【請求項１】　固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いか否かを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切換えるスケジューリングモード切換手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てることを特徴とする計算機システム。

【請求項２】　固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段と、エラー発生等の非周期的イベントの発生を上記スケジューリング実行手段に割込みで通知する割込み発生手段と、上記割込み発生時の処理を行なう割込み処理手段とを備え、上記スケジューリング実行手段は、上記割込み発生時、予め定められた上記周期的タスクと割込み処理の優先度に基づき割込み処理が周期的タスクより優先度が高い時は、上記周期的タスクの実行を一時中断して上記割込み処理手段に割込み処理を実行させることを特徴とする計算機システム。

【請求項３】　固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いか否かを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切換えるスケジューリングモード切換手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段と、ハードウェアタイマからのクロックに基づきタイムスロット時間毎に上記スケジューリング実行手段に割込みを発生する割込み発生器と、ハードウェアタイマからのクロックに基づき各タイムスロット値をカウントするカウンタと、上記カウンタによりタイムスロット値がセットされ、上記割込み発生器からの割込み発生毎に上記スケジューリング実行手段により現時刻のタイムスロット値が読み出されるレジスタとを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てることを特徴とする計算機システム。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】この発明は、複数タスクが実行される計算機システムに係わり、特に固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムに関するものである。

【０００２】

【従来の技術】図１５は、例えば「三菱電機統合制御システムＭＥＬＴＡＳ　ＪＰ－ＥＩ　Ｓ／Ｗ構築マニュアル」（１９９１年５月第１版発行）で提供されている計算機システムのスケジューリングに関する部分の構成図である。

【０００３】まず、周期的スケジューリングの概念について、図１６を用いて説明する。図中の最小基準周期とは、周期的にタスクの起動をチェックする際の基準周期となるもので、周期的に実行されるアプリケーションタ

2

スク１～Ｌ（Ｌはタスク数）をスケジューリングする最小周期である。タイムスロットとは、上記最小基準周期を所定値Ｎ（ユーザがシステムに合わせて決定する）で割った時間単位で、タイムスロット時間毎に、スケジューラが実行タスク１～Ｌの選択を行う。実行レベルとは、実行タスク１～Ｌを周期毎に分類した複数のタスクの集合で、タイムスロット１～Ｎにその実行レベル１～Ｍが割込てられる（割当て方法は後述する）。スケジューラはスケジューリング実行時に、あるタイムスロットでそのタイムスロットに割当てられた該実行レベルの起動待ちタスクを優先的に実行する（実行レベルの選択方法は後述する）。

【０００４】図１６の例では、タイムスロット１とタイムスロット２に実行レベル１を、タイムスロット３に実行レベル２を、タイムスロット（Ｎ－１）に実行レベルＭを割当て、タイムスロットＮは、実行レベルの選択時間やタイムスロットの計算時間として、各タイムスロットに要するシステム時間のための予備のタイムスロット（α）であり、実行レベル１を割当てる。実行レベル１にはタスク１とタスク２の２つのタスクが、実行レベル２にはタスク３が、実行レベルＭにはタスクＬが所属している。スケジューリング時、タイマの値から、後述するタイムスロット計算部がタイムスロット値を計算する。タイムスロット１においては、実行レベル１が選択され、実行レベル１のタスク１が実行されている。タイムスロット２においても、実行レベル１が選択され、同様にタスク１が実行されている。タイムスロット３においては実行レベル２が選択され、実行レベル２の所属タスク３が実行される。タイムスロットＮは、予備のタイムスロットなので、最小基準周期内で未実行のタスクが実行される。以上の様な実行レベル選択が、最小基準周期毎に繰り返される。

【０００５】図１５において、１１は、ユーザから提供される各タスクに対する固有の実行周期と実行時間を含むタスク情報１２からタスクを実行レベルに分類し、実行レベルを割当てたタイムスロットの情報を含むテーブル情報（変更値）２３を生成し、タイムスロット対応タスク情報テーブル２１に該テーブル情報２３を格納するタイムスロット割当て部であり、図１７に示すように、タスク情報１２から各タスクの実行周期を判別する実行周期判別手段１１ａと、この判別結果に基づき各タスクを実行周期の短い実行レベルから長いレベルまでレベル分けする分類手段１１ｂと、各タイムスロットに上記実行レベルを割当てるタイムスロット割当て手段１１ｃ等から成る。一方、図１５の１３は後述するタイムスロット計算部１６からのタイムスロット変更割込み１７をトリガとして、タイムスロット対応タスク情報テーブル２１からのテーブル情報（現在値）２２と、後述するタイムスロット情報格納部１８からのタイムスロット値（現在値）２０に基づき実行タスクを選択し、起動をかけるスケジューリング実行部、１６はオペレーティングシステムの機能であるソフトウエア（以下、Ｓ／Ｗと略記する）タイマ１４からの定周期割込み１５で起動され、現在の時刻からタイムスロット値（変更値）１９を算出し、タイムスロット情報格納部１８にその値を書き込むとともに、このタイムスロット情報変更時にスケジューリング実行部１３にタイムスロット変更割込み１７をかけ、現在のタスク実行を中断させ、起動タスク選択をスケジューリング実行部１３に行わせるタイムスロット計算部であり、算出したタイムスロット値は内部にも保持され、このタイムスロット値が次回の算出時にタイムスロットの前回値としてインクリメントされる。２４は、ユーザからタスク情報１２として渡される起動要求タスク２５を、タイムスロット対応タスク情報テーブル２１内の後述するタスク管理テーブルで生成されるキュー（待ち行列）にキューイングする起動タスクキューイング部である。

【０００６】図１８は図１５のタイムスロット対応タスク情報テーブル２１に含まれる各テーブルの詳細図である。２１１はタイムスロット１～Ｎの各々に割当てた実行レベル１～Ｍの情報を格納するタイムスロット管理テーブル、２１２ａ～２１２ｃは各タイムスロット値に割当てられた各実行レベルのタスクのキューイング状態を格納するレベル管理テーブル、２１３ａ１～２１３ａｎ，２１３ｂ１，２１３ｃ１，２１３ｃ２は上述したキューを生成するためのタスク管理テーブルである。上記タイムスロット管理テーブル２１１には、各タイムスロット１～Ｎに割当てられた実行レベル１～Ｍに対応するレベル管理テーブル２１２ａ～２１２ｃのアドレスを示すポインタａ～ｃが格納される。タイムスロットＮはシステム時間のための予備のタイムスロット（α）で、実行レベル小のタスクを優先的に実行するために実行レベル１を割当てる。タスク管理テーブル２１３ａ１～２１３ｃ２には、次のタスク管理テーブルのアドレスを指すポインタ（ただし、キューの最後尾のタスク管理テーブル２１３ａｎ，２１３ｂ１，２１３ｃ２のポインタは後に続くテーブルがないことを示すＮＵＬＬポインタ）と、起動タスクの情報が格納される。また、レベル管理テーブル２１２ａ～２１２ｃには、各々の実行レベルでのキューの先頭にあるタスク管理テーブルのアドレスを指すポインタと、最後尾にあるタスク管理テーブルのアドレスを指すポインタが格納される。例えば、レベル１用のレベル管理テーブル２１２ａには、実行レベル１に所属するキューイングタスクの先頭タスク管理テーブル２１３ａ１のアドレスを示すポインタａ１と、同最後尾タスク管理テーブル２１３ａｎのアドレスを示すポインタａｎを格納することで、これらのタスク管理テーブル２１３ａ１～２１３ａｎに示されたタスクが実行待ちタスクであるという情報を提供する。

【０００７】次に動作について説明する。まず、タイム

スロット割当て部１１による上記タイムスロット管理テーブル２１１への実行レベルの割当て手段について、図１９，図２０のフロー図を用いて説明する。割当てに先立って、入力されたタスク情報１２に基づき、実行周期判別手段１１ａと分類手段１１ｂにより、全ての起動タスクを実行周期毎にM個（ユーザが実行タスク数とシステムの能力から決定する数）の実行レベルに分類する（ＳＴ１９１）。次に、タイムスロット割当て手段１１ｃにより、実行レベル１～Mについて以下の処理を繰り返す（ＳＴ１９２）。まず、実行レベル１について、最小基準周期単位の実行時間の合計を、実行レベル１に所属する全てのタスク実行時間の総和に、実行レベル１の実行周期の最小基準周期に対する割合を掛けることで求める（ＳＴ１９３）。例えば、実行レベル１のタスクの実行周期が１００ｍｓ、実行レベル１のタスクの実行時間合計が４０ｍｓ、最小基準周期が５０ｍｓの時、４０×（５０／１００）＝２０ｍｓが求める値となる。即ち、実行レベル１は最小基準周期内で２０ｍｓの実行時間が必要となる。その結果をタイムスロット時間で割り、小数点以下を切上げ、最小基準周期内で実行レベル１に必要なタイムスロット数を求める（ＳＴ１９４）。前述の例では、タイムスロット時間が１０ｍｓの時、２０／１０＝２、故に２個のタイムスロットが必要となる。以上の処理を同様に実行レベル２からMまで繰り返し（ＳＴ１９５）、各実行レベルに必要なタイムスロット数を求める。次に最小基準周期内でタスクの実行に必要となるシステムの実行時間に必要なタイムスロット数を、システム実行時間をタイムスロット時間で割り、少数点以下を切り上げて求める（ＳＴ１９６）。以上で、各々の実行レベルとシステムが必要とするタイムスロット数が算出できるが、ただし、その合計が最小基準周期のタイムスロット数Ｎを越えると、実行最大負荷を越えスケジューリング不可となるため、その合計と最小基準周期のタイムスロット数Ｎを比較し（ＳＴ１９７）、合計の方がタイムスロット数Ｎより大きい場合は、実行レベル１から順に必要タイムスロット数だけ、レベル管理テーブルへのポインタをタイムスロット管理テーブル２１１に格納し、タイムスロット管理テーブル２１１が満たされたとき、残りのレベルは無視するようにする（ＳＴ１９９）。合計がタイムスロット数Ｎより小さい場合は、実行レベル１から順にタイムスロット管理テーブル２１１に格納する（ＳＴ１９８）。以上でタイムスロットへの実行レベルの割当てを終了する。

【０００８】次に、スケジューリング実行部１３によるタイムスロット管理テーブル２１１を用いたスケジューリング方法を、図２１のフロー図を用いて説明する。Ｓ／Ｗタイマ１４からの定周期タイマ割込み１５を受けて、タイムスロット計算部１６はタイムスロット値１９を計算し、タイムスロット情報格納部１８にその値を格納し、タイムスロット変更割込み１７をスケジューリング実行部１３に出力する。スケジューリング実行部１３はこの変更割込み１７を受けて、動作を開始する。現在のタイムスロット値２０をタイムスロット情報格納部１８から獲得し（ＳＴ２１１）、獲得したタイムスロット値に割当てられている実行レベルを図１８のタイムスロット管理テーブル２１１の実行レベル１～Mから獲得する（ＳＴ２１２）。該実行レベルにキューイングされたタスク（実行待ちタスク）の存在を、該レベル管理テーブル２１２ａ～２１２ｃを用いてチェックし（ＳＴ２１３）、キューイングされていればその先頭タスクを実行する（ＳＴ２１４）。例えば、タイムスロット情報格納部１８からのタイムスロット値が１の時、スケジューリング実行部１３はタイムスロット管理テーブル２１１によりポインタａで示されるレベル管理テーブル２１２ａを確認し、そのポインタａ１で示される先頭タスク管理テーブル２１３ａ１のタスクを実行する。キューインイグされていなければ実行レベルをインクリメントし（ＳＴ２１６）、次の実行レベルのレベル管理テーブルについてキューイング状態のチェックを繰り返す。実行レベルのチェックは、現タイムスロットに割り当てられた実行レベルを先頭に、図２２のようにサイクリックに行うため、実行レベルがMを越えたら実行レベルを１に戻す（ＳＴ２１７，ＳＴ２１８）。以上のチェックで全ての実行レベルを一巡したら（ＳＴ２１９）、全ての実行レベルに実行タスク無しと判断する（ＳＴ２２０）。実行タスクが有る場合，無い場合ともに、タイムスロット変更割込み１７を待って（ＳＴ２１５）、次のタイムスロット値の獲得へ戻り、以上の動作を繰り返す。

【０００９】

【発明が解決しようとする課題】従来の計算機システムは以上の様に構成されているので、図１９，図２０に示したタイムスロット割当て方法では、周期的スケジューリングを行なうことでプロセッサへの負担を均等にできるが、最小基準周期でタスク起動の周期的チェックを実行するため、最小基準周期未満の周期的スケジューリングは不可能となる。このため、最小基準周期未満の実行周期のタスクについては、このようなタスクが存在しないようにユーザは実行周期を決定せねばならないという問題点があった。また、予めユーザより指定された実行要求タスク，即ちタイムスロット管理テーブルに割付けられたタスクしか実行できないので、周期的タスクとともに、エラー処理等の非周期的に発生する割込み処理を行なうことができないという問題点があった。

【００１０】この発明は上記のような課題を解決するためになされたものであり、従来の周期的スケジューリングに加え、実行周期が最小基準周期未満のタスクが存在する場合のスケジューリングを提供し、全ての周期のタスクのスケジューリングを可能にした計算機システムを得ることを目的とする。また、周期的タスクとともに、エラー処理等の非周期的に発生する割込み処理も可能に

した計算機システムを得ることを目的とする。

【００１１】

【課題を解決するための手段】この発明の第１の発明に係る計算機システムは、固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いか否かを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切換えるスケジューリングモード切換手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるようにしたものである。

【００１２】また、第２の発明に係る計算機システムは、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備えるとともに、エラー発生等の非周期的イベントの発生を上記スケジューリング実行手段に割込みで通知する割込み発生手段と、上記割込み発生時の処理を行なう割込み処理手段とを備え、上記スケジューリング実行手段は、上記割込み発生時、予め定められた上記周期的タスクと割込み処理の優先度，即ちプライオリティに基づき割込み処理が周期的タスクよりプライオリティが高い時は、上記周期的タスクの実行を一時中断して上記割込み処理手段に割込み処理を実行させるようにしたものである。

【００１３】一方、第３の発明に係る計算機システムは、上記第１の発明同様、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いか否かを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切換えるスケジューリングモード切換手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備えるとともに、ハードウェアタイマからのクロックに基づきタイムスロット時間毎に上記スケジューリング実行手段に割込みを発生する割込み発生器と、ハードウェアタイマからのクロックに基づき各タイムスロット値をカウントするカウンタと、上記カウンタによりタイムスロット値がセットされ、上記割込み発生器からの割込み発生毎に上記スケジューリング実行手段により現時刻のタイムスロット値が読み出されるレジスタとを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるようにしたものである。

【００１４】

【作用】第１の発明における計算機システムは、タイムスロット割当て手段が実行レベルのタイムスロット割当てを実行するに先立って、上記実行周期判別手段，分類手段及び比較手段により、最小基準周期未満の実行周期のタスクの存在をチェックする。もし存在した場合は、スケジューリングモード切換手段が通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換える。タイムスロット割当て手段は、スケジューリングモードがプライオリティスケジューリングモードとなっていることを認識すると、全てのタイムスロットに最小実行周期の実行レベルを割当てる。スケジューリング実行手段は、全てのタイムスロットで、割当てられた最小実行周期の実行レベルから順にタスクを実行する。全てのタイムスロットに最小実行周期の実行レベルを割当てることで、常に最小実行周期の実行レベル

から順にタスクをチェックすることになり、結果として、最小実行周期の実行レベルに最大のプライオリティを与え、タイムスロット単位でタスクの起動をチェックするプライオリティスケジューリングが実現される。

【００１５】また、第２の発明における計算機システムは、スケジューリング実行手段が割込み発生手段からの割込みの有無をチェックし、割込みが無い場合は通常の周期的スケジューリングを行なう。割込みが有れば、周期的タスクと割込み処理のプライオリティに基づき、割込み処理のプライオリティが高ければ割込み処理手段に割込み処理を行なわせる。割込み処理のプライオリティが低ければ通常の周期的スケジューリングを行なう。これにより、周期的スケジューリングと、エラー処理等の非周期的に発生する割込み処理のプライオリティ制御が可能となる。

【００１６】一方、第３の発明における計算機システムは、ハードウェアタイマからのクロックにより割込み発生器及びカウンタが動作し、割込み発生器からスケジューリング実行手段へタイムスロット時間毎に割込みが発生するとともに、カウンタからレジスタに各時刻のタイムスロット値がセットされる。スケジューリング実行手段は、タイムスロット時間毎の割込みが入るとレジスタを見に行き、現時刻に対応するタイムスロット値を得る。このタイムスロット値を用いて、上記第１の発明と同様のスケジューリング処理を行なう。タイマにハードウェアタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる。

【００１７】

【実施例】実施例１．以下、この発明の実施例を図について説明する。図１は、実施例１による計算機システムのスケジューリングに関する部分の構成図である。図において、２６は新たに設けられたスケジューリングモード選択部であり、タスク情報１２から、起動タスクの実行周期中，最小基準周期未満のタスクの有無をチェックし、もし存在すればスケジューリングモードとしてプライオリティスケジューリングモードを選択し、存在しなければ従来の周期的スケジューリングモードを選択する。２７は上記スケジューリングモード選択部２６が選択結果として出力するスケジューリングモード信号である。１１は、上記スケジューリングモード選択部２６からスケジューリングモード信号１７を受け取り、周期的スケジューリングモードなら従来のものと同様の動作を行い、プライオリティスケジューリングモードなら、全てのタイムスロットに実行レベル１を割当てるタイムスロット割当て部であり、この機能は図１７に示したタイムスロット割当て手段１１ｃにより実現される。その他については従来のシステムと同様である。

【００１８】図２は上記スケジューリングモード選択部２６の内部構成を示す機能ブロック図である。図において、２６ａはタスク情報１２に基づき各タスクの実行周期を判別する実行周期判別手段、２６ｂは上記実行周期判別手段２６ａの判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段、２６ｃは上記分類手段２６ｂにより最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いか否かを判定する比較手段、２６ｄはこの比較結果に基づき、従来からの周期的スケジューリングモードと、新たなスケジューリングモードとして全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切換えるスケジューリングモード切換手段であり、このスケジューリングモード切換手段２６ｄの出力がスケジューリングモード信号２７となる。なお、上記実行周期判別手段２６ａと分類手段２６ｂは、従来からタイムスロット割当て部１１に備わっている実行周期判別手段１１ａ及び分類手段１１ｂ（図１７参照）と同様なものである。従って、これらを一組だけ備えてタイムスロット割当て部１１とスケジューリングモード選択部２６で共用することも可能である。

【００１９】次に、本実施例におけるタイムスロット割当て方法について図３，図４のフロー図を用いて説明する。まず、スケジューリングモード選択部２６は、実行周期判別手段２６ａにより全ての起動タスクの実行周期を判別し、分類手段２６ｂにより実行周期別に実行レベルに分類する（ＳＴ３１）。例えば、実行周期５０ｍｓのタスクは実行レベル１に、実行周期７０ｍｓのタスクは実行レベル２に、という具合に分類する。次に、スケジューリングモード選択部２６は、比較手段２６ｃにより最小実行周期のタスクが所属する実行レベル１の実行周期と最小基準周期を比較し（ＳＴ３２）、実行レベル１の周期の方が小さければ、スケジューリングモード切換手段２６ｄによりプライオリティスケジューリングモードを選択し（ＳＴ３４）、大きければ従来同様の周期的スケジューリングモードを選択する。（ＳＴ３３）。次に、タイムスロット割当て部１１は、スケジューリングモード信号２７を受け取り、このモードをチェックし（ＳＴ３５）、これがプライオリティスケジューリングモードの場合は、図５に示すように、タイムスロット管理テーブル２１１の全てのタイムスロット１～Ｎに実行レベル１を割当てる（ＳＴ３６）。スケジューリングモードが周期的スケジューリングモードの場合は、従来のシステムと同様の処理を行なう。即ち、タイムスロット割当て部１１は、実行レベル１～Ｍについて以下の処理を行う（ＳＴ３７）。実行レベル１について、実行レベル１の全てのタスク実行時間の総和に、実行レベル１の実行周期の最小基準周期に対する割合を掛けることで、実行レベル１のタスクの実行時間合計が最小基準周期内に占める時間を求める（ＳＴ３８）。従来例でも述べたように、例えば、実行レベル１のタスクの実行周期が１

００ｍｓ、実行レベル１のタスクの実行時間合計が４０ｍｓ、最小基準周期が５０ｍｓの時、４０×（５０／１００）＝２０ｍｓが求める値となる。即ち、実行レベル１は最小基準周期内で２０ｍｓの実行時間が必要となる。その結果をタイムスロット時間で割り、小数点以下を切上げ、最小基準周期内で実行レベル１に必要なタイムスロット数を求める（ＳＴ３９）。前述の例では、タイムスロット時間が１０ｍｓの時、２０／１０＝２、故に２個のタイムスロットが必要となる。以上の処理を同様に実行レベル２からＭまで繰り返し（ＳＴ４０）、各実行レベルに必要なタイムスロット数を求める。次にタスクの実行に必要となるシステムの実行時間に必要なタイムスロット数を求める（ＳＴ４１）。以上で、各々の実行レベルが必要とするタイムスロット数が算出できるが、ただし、その合計が最小基準周期のタイムスロット数Ｎを越えると、実行最大負荷を越え制御不可となるため、その合計とＮを比較し（ＳＴ４２）、合計がタイムスロット数Ｎを越えている場合は、実行レベル１から順に必要タイムスロット数だけタイムスロット管理テーブル２１１に格納し、テーブルが満たされたとき、残りのレベルは無視するようにする（ＳＴ４４）。Ｎのほうが大きい場合は問題なく、実行レベル１から順にタイムスロット管理テーブル２１１に格納する（ＳＴ４３）。以上でタイムスロットへの実行レベルの割当てを終了する。

【００２０】タイムスロット割当て後のスケジューリング実行は、従来の手段と同様の動作を行う。この時、プライオリティスケジューリングモードで図５に示したようにタイムスロット管理テーブル２１１が生成された場合は、従来例で示した図２１のフロー図は図６に示すようになり、全てのタイムスロットで実行レベル１を得て（ＳＴ２１２）、実行レベル１から図２２に従ってレベル管理テーブル２１２ａ～２１２ｃをチェックし、最初に発見したキューイングタスクを実行する。この結果、実行レベル１に最大のプライオリティを与え、実行レベルＭに最小のプライオリティを与えることになる。即ち、どのタイムスロットにおいても実行レベル１を獲得し、図２２の実行レベル探索に従い実行タスクを決定するので、実行レベルの小さいタスクが終了しない限り、大きいタスクは実行されない。この結果、実行レベルの小さいタスクほど高いプライオリティを与えるプライオリティスケジューリングが実現される。この場合、従来例で示した図１６は図７に示すようになる。

【００２１】実施例２．図８は実施例２の計算機システムのスケジューリングに関する部分の構成図である。本実施例は、図１の実施例１のものに、エラー発生等の非周期的イベントの発生をスケジューリング実行部１３に割込みで知らせる割込み発生部２８と、割込みが生じた時の処理を行なう割込み処理部２９と、予め定められた周期的タスクと上記割込み処理との間のプライオリティ情報を格納しておくプライオリティ情報格納部３０とを設けたものである。そして、スケジューリング実行部１３は、通常は実施例１のような周期的スケジューリングを行い、エラー発生等により割込み発生部２８より割込み（割込み種類を含む）３１が入った時は、プライオリティ情報格納部３０からのプライオリティ情報３２に基づき、その割込み処理が周期的タスクよりプライオリティが高ければ、割込み処理要求（割込み種類を含む）３３を割込み処理部２９へ送り、周期的タスクの実行を一時中断して割込み処理を行なうようにしたものである。

【００２２】上記割込み発生部２８は、図９に示すように、割込み種類通知テーブル２８ａと、割込み発生通知テーブル２８ｂと、実行タスク監視部２８ｃとを有しており、以下、エラー割込みを例にとって説明する。計算機システムの実行中タスク２８ｄは自タスク内に何らかのエラーが生じると、その割込み種類２８ｅを割込み種類テーブル２８ａに書き込み、割込み発生通知テーブル２８ｂには割込み発生信号２８ｆをセットする。一方、実行タスク監視部２８ｃは、割込み発生通知テーブル２８ｂを一定周期でポーリングし、該テーブル２８ｂからの割込み発生信号２８ｇがセットされると、割込み種類通知テーブル２８ａから割込み種類２８ｈを獲得し、それを割込み３１によりスケジューリング実行部１３へ送る。処理終了後は、再び割込み発生通知テーブル２８ｂのポーリングに入る。一方、上記割込み処理部２９は、図１０に示すように、割込み種類通知テーブル２９ａと、割込み処理要求通知テーブル２９ｂと、割込みハンドラ２９ｃと、割込み種類に対応した複数の割込み処理ルーチン（１）２９ｄ～（ｎ）２９ｆとを有している。スケジューリング実行部１３は、割込み発生部２８から割込み種類を含む割込み３１を受けとると、割込み種類に対応する割込み処理のプライオリティと周期的タスクのプライオリティを比較し、割込み処理のプライオリティの方が高ければ、割込み処理部２９の割込み種類通知テーブル２９ａに該割込み種類３３ｂを書き込み、その後，割込み処理要求通知テーブル２９ｂに割込み処理要求３３ａをセットする。一方、割込みハンドラ２９ｃは、割込み処理要求通知テーブル２９ｂを一定周期でポーリングし、もし該テーブル２９ｂからの割込み処理要求２９ｇがセットされると、割込み種類通知テーブル２９ａから割込み種類２９ｈを獲得し、それに応じて割込み処理起動信号２９ｉ～２９ｋを出力し割込み処理ルーチン（１）２９ｄ～（ｎ）２９ｆのうち１つを起動する。処理が終了すると、再び割込み処理要求通知テーブル２９ｂのポーリングに入る。

【００２３】次に、本実施例のスケジューリング動作について、図１１のフロー図を用いて説明するまず、スケジューリング実行部１３が割込みの有無をチェックし（ＳＴ１１１）、割込みが無い場合は実施例１同様の周期的スケジューリングを行なう。割込みが有れば以下の

処理を行なう。割込み処理と周期的タスクの間のプライオリティが予め格納されたプライオリティ情報格納部３０から、割込み処理と周期的タスクのプライオリティ情報３２を得てこれらを比較し（ＳＴ１１２）、割込み処理のプライオリティが高ければ、割込み処理を実行し（ＳＴ１１４）、割込みをクリアする（ＳＴ１１５）。割込み処理のプライオリティが低ければ、割込みをペンディングし（ＳＴ１１３）、実施例１における通常の周期的スケジューリングを行なう。ただし、この周期的スケジューリングにおいて、実行タスクが無い場合（ＳＴ１２４）、このタイムスロットを利用してペンディング割込みの処理を行なう（ＳＴ１２５→ＳＴ１２６）。割込み処理後は割込みをクリアする（ＳＴ１２７）。以上の動作により、周期的スケジューリングと、エラー処理等の非周期的に発生する割込み処理のプライオリティ制御も可能となる。即ち、従来例や実施例１においては、予めユーザにより指定されてタイムスロット管理テーブル２１１に割付けられたタスクしか実行できないが、上記のような割込み発生部２８と割込み処理部２９を付加することで、非周期的に発生するエラー処理などを、そのプライオリティを予め高くしておくことにより、周期的スケジューリングを一時中断して実行することができる。なお、本実施例では、図１の実施例１のものに上記のような割込み処理機能を付加したものについて説明したが、図１５の従来例のものに適用することも可能である。

【００２４】実施例３．図１３は実施例３の計算機システムのスケジューリングに関する部分の構成図である。本実施例は、図１の実施例１におけるＳ／Ｗタイマ１４，タイムスロット計算部１６及びタイムスロット情報格納部１８をハードウェアで構成したものである。即ち、３４は図１のＳ／Ｗタイマ１４の代りとなるハードウェア（以下、Ｈ／Ｗと略記する）タイマで、計算機システムが元々有するものである。３５は上記Ｈ／Ｗタイマ３４からのクロック３６に基づきタイムスロット時間毎にスケジューリング実行部１３に割込み３７を発生する割込み発生器、３８～４０は上記Ｈ／Ｗタイマ３４からのクロック３６に基づき各タイムスロット値１～Ｎをそれぞれカウントする Ｎ個のカウンタで、これら割込み発生器３５及びカウンタ３８～４０は図１のタイムスロット計算部１６の代りに設けられたものである。４１は上記各カウンタ３８～４０の出力４２により各タイムスロット値１～Ｎに対応するビットがセットされ、上記割込み発生器３５からの割込み発生毎にスケジューリング実行部１３により現時刻のタイムスロット値４３が読み出されるタイムスロットレジスタで、これは図１のタイムスロット情報格納部１８の代りに設けられたものである。

【００２５】次に、本実施例のスケジューリング動作について、図１４のフロー図を用いて説明する。Ｈ／Ｗタイマ３４のクロック３６を用い、割込み発生器３５はタイムスロット時間毎にスケジューリング実行部１３に割込み３７を発生する。カウンタ（１）３８からカウンタ（Ｎ）４０はＨ／Ｗタイマ３４のクロック３６を利用して、それぞれ定められたタイムスロット値１～Ｎをカウントし、最小基準周期毎にタイムスロットレジスタ４１の対応ビットをセットする。スケジューリング実行部１３は、割込み発生器３５からタイムスロット時間毎の割込み３７が入るとタイムスロットレジスタ４１を見に行き、セットされたビットからタイムスロット値４３を得る（ＳＴ１４１）。このタイムスロット値を用いて、実施例１と同様のスケジューリング処理を行ない、Ｈ／Ｗタイマ割込みを待って（ＳＴ１４５）、次のタイムスロット値の獲得へ戻り、以上の動作を繰り返す。タイマにＨ／Ｗタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる。即ち、実施例１では、オペレーティングシステムのＳ／Ｗタイマ機能を用い、現在の時刻を獲得する処理，タイムスロットの計算，スケジューリング実行部１３へのタイムスロット変更割込みの通知といった処理を全てソフトウエアで実現している。この場合、タイムスロットの前回値の保持が必要な上、オーバヘッド時間（図７のシステム時間）が大きくなるという欠点がある。これに対し、本実施例のように、以上の機能を、Ｈ／Ｗタイマを用いて割込み発生器，カウンタ及びレジスタなどのハードウェアで実現した場合、タイムスロットの前回値保持が不要となる上、オーバヘッドを低減でき、高速なスケジューリングが可能となる。

【００２６】

【発明の効果】以上のように、この発明の第１の発明によれば、固有の実行周期と実行時間を保有する複数のタスクを管理してスケジューリングする機能を有する計算機システムにおいて、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いか否かを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切換えるスケジューリングモード切換手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段

とを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるようにしたので、可能な限り周期的スケジューリングを行うことでプロセッサへの負担を均等にするとともに、スケジューリング不可能な周期のタスクが存在する場合は、それを最優先で実行することになり、全ての周期のタスクに対応したスケジューリングが可能になるという効果が得られる。

【００２７】また、第２の発明によれば、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備えるとともに、エラー発生等の非周期的イベントの発生を上記スケジューリング実行手段に割込みで通知する割込み発生手段と、上記割込み発生時の処理を行なう割込み処理手段とを備え、上記スケジューリング実行手段は、上記割込み発生時、予め定められた上記周期的タスクと割込み処理のプライオリティに基づき割込み処理が周期的タスクよりプライオリティが高い時は、上記周期的タスクの実行を一時中断して上記割込み処理手段に割込み処理を実行させるようにしたので、周期的スケジューリングと、エラー処理等の非周期的に発生する割込み処理のプライオリティ制御が可能となる効果が得られる。

【００２８】一方、第３の発明によれば、上記第１の発明同様に、各タスクの実行周期を判別する実行周期判別手段と、この実行周期判別手段の判別結果に基づき、各タスクを実行周期の短い実行レベルから長い実行レベルまでレベル分けする分類手段と、周期的にタスクの起動をチェックする際の基準となる最小基準周期を所定値で割った時間単位であるタイムスロットに、上記実行レベルを割当てるタイムスロット割当て手段と、最小実行周期のタスクが分類された実行レベルの実行周期が上記最小基準周期より短いか否かを判定する比較手段と、この比較結果に基づき、各タイムスロットに予め設定された周期に従って上記各実行レベルを割当てる周期的スケジューリングモードと、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるプライオリティスケジューリングモードとを切換えるスケジューリングモード切換手段と、現時刻に対応するタイムスロットに割当てられた実行レベルからタスクスケジューリングを実行するスケジューリング実行手段とを備えるとともに、ハードウェアタイマからのクロックに基づきタイムスロット時間毎に上記スケジューリング実行手段に割込みを発生する割込み発生器と、ハードウェアタイマからのクロックに基づき各タイムスロット値をカウントするカウンタと、上記カウンタによりタイムスロット値がセットされ、上記割込み発生器からの割込み発生毎に上記スケジューリング実行手段により現時刻のタイムスロット値が読み出されるレジスタとを備え、上記比較手段で最小実行周期のタスクが分類された実行レベルの実行周期が最小基準周期よりも短いと判定されたとき、上記スケジューリングモード切換手段は通常の周期的スケジューリングモードからプライオリティスケジューリングモードに切換え、これに基づき上記タイムスロット割当て手段は、全てのタイムスロットに最小実行周期のタスクが分類された実行レベルを割当てるようにしたので、上記第１の発明と同様な効果が得られるとともに、タイマにH／Wタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる効果が得られる。

【図面の簡単な説明】

【図１】この発明の実施例１による計算機システムのスケジューリングに関する部分の構成図である。

【図２】図１のスケジューリングモード選択部の内部構成を示す機能ブロック図である。

【図３】図１のスケジューリングモード選択部とタイムスロット割当て部の動作を示すフロー図である。

【図４】図３の続きを示すフロー図である。

【図５】プライオリティスケジューリング時のタイムスロット対応タスク情報テーブル内の詳細図である。

【図６】プライオリティスケジューリング時のスケジューリング実行部の動作を示すフロー図である。

【図７】プライオリティスケジューリング時のタスクの実行状態を示す図である。

【図８】この発明の実施例２による計算機システムのスケジューリングに関する部分の構成図である。

【図９】図８の割込み発生部の内部構成図である。

【図１０】図８の割込み処理部の内部構成図である。

【図１１】図８のスケジューリング動作を示すフロー図である。

【図１２】図１１の続きを示すフロー図である。

【図１３】この発明の実施例３による計算機システムのスケジューリングに関する部分の構成図である。

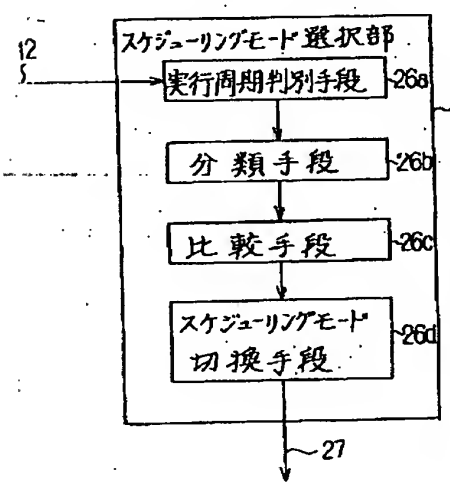
【図１４】図１３のスケジューリング動作を示すフロー図である。

【図１５】従来の計算機システムのスケジューリングに関する部分の構成図である。

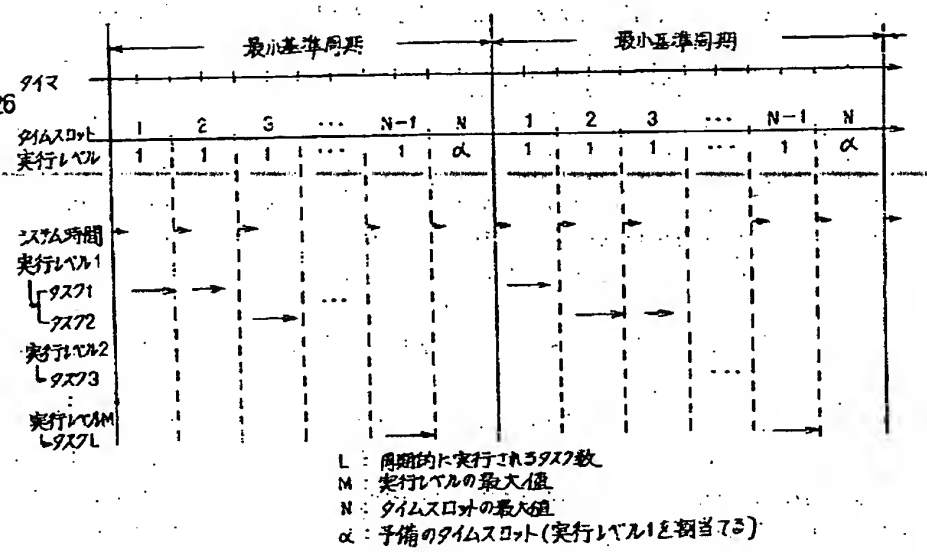
【図１６】スケジューリング時のタスクの実行状態を示

す図である。
【図１７】タイムスロット割当て部の内部構成を示す機能ブロック図である。
【図１８】タイムスロット対応タスク情報テーブル内の詳細図である。
【図１９】図１５のタイムスロット割当て部の動作を示すフロー図である。
【図２０】図１９の続きを示すフロー図である。
【図２１】スケジューリング実行部の動作を示すフロー図である。
【図２２】実行レベルのチェックの順番を示す図である。
【符号の説明】
１１　タイムスロット割当て部
１１ａ　実行周期判別手段
１１ｂ　分類手段
１１ｃ　タイムスロット割当て手段
１２　タスク情報
１３　スケジューリング実行部（スケジューリング実行手段）
１４　Ｓ／Ｗタイマ
１５　定周期割込み
１６　タイムスロット計算部
１７　タイムスロット変更割込み
１８　タイムスロット情報格納部
１９　タイムスロット値（変更値）
２０　タイムスロット値（現在値）
２１　タイムスロット対応タスク情報テーブル
＊２２　テーブル情報（現在値）
２３　テーブル情報（変更値）
２４　起動タスクキューイング部
２５　起動要求タスク
２６　スケジューリングモード選択部
２６ａ　実行周期判別手段
２６ｂ　分類手段
２６ｃ　比較手段
２６ｄ　スケジューリングモード切換手段
２７　スケジューリングモード信号
２８　割込み発生部（割込み発生手段）
２９　割込み処理部（割込み処理手段）
３０　プライオリティ情報格納部
３１　割込み（割込み種類を含む）
３２　プライオリティ情報
３３　割込み処理要求（割込み種類を含む）
３４　Ｈ／Ｗタイマ
３５　割込み発生器
３６　クロック
３７　割込み
３８～４０　カウンタ
４１　タイムスロットレジスタ
４２　カウンタ出力
４３　タイムスロット値
２１１　タイムスロット管理テーブル
２１２ａ～２１２ｃ　レベル管理テーブル
２１３ａ１～２１３ａｎ，２１３ｂ１，２１３ｃ１，２１３ｃ２　タスク管理テーブル

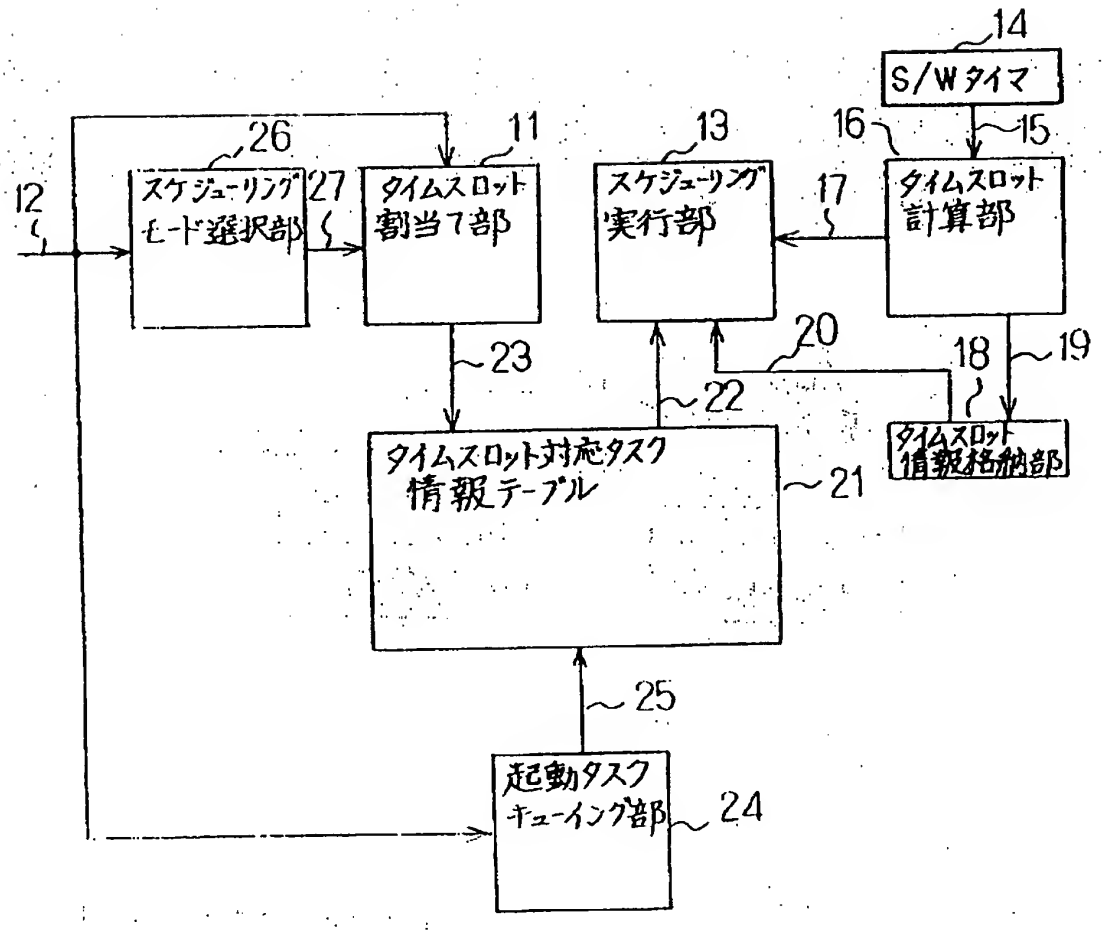
【図２】



【図７】

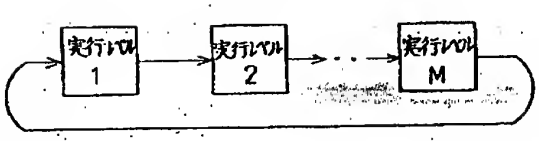


【図1】

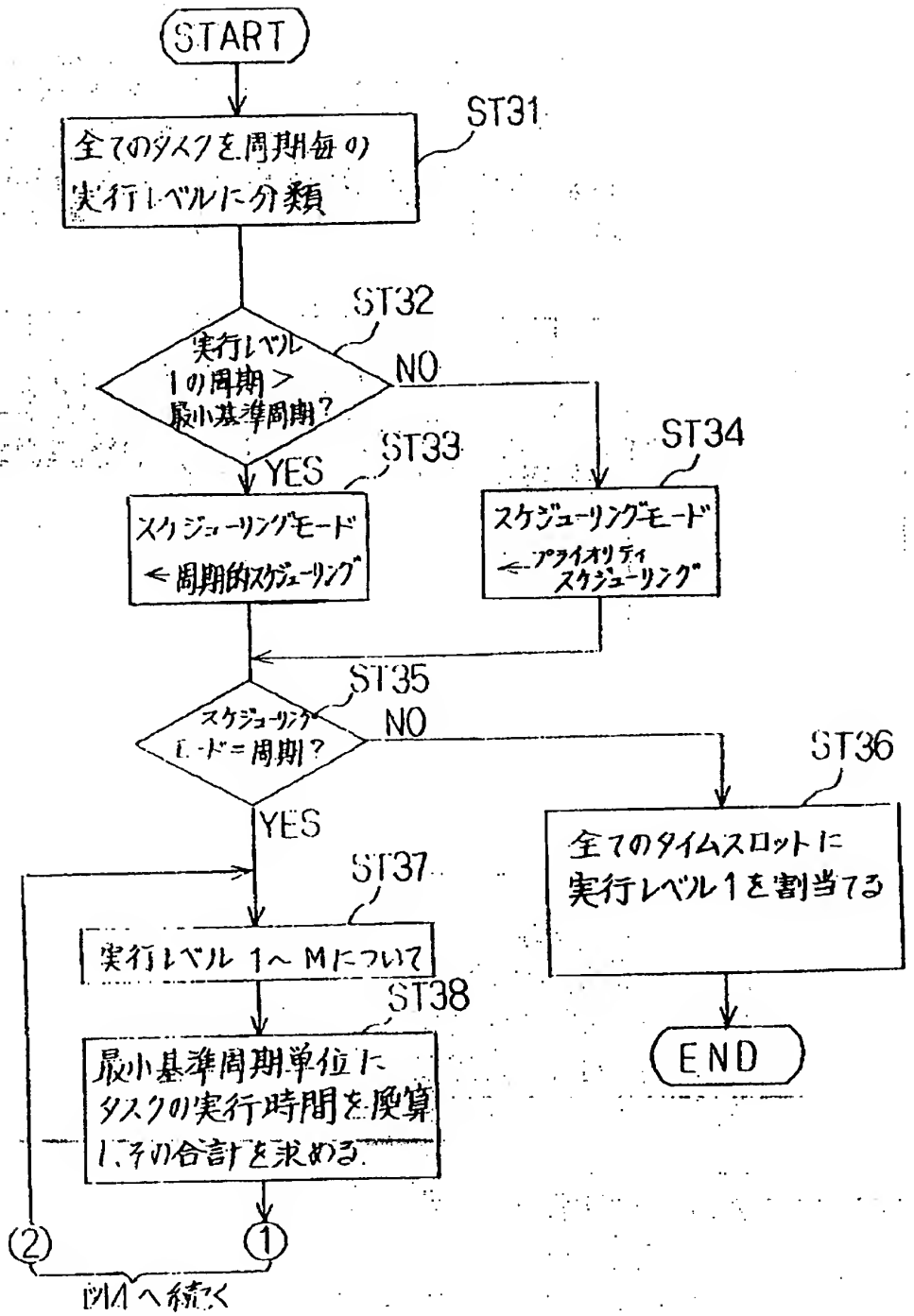


12 ; タスク情報
15 ; 定周期割込み
17 ; タイムスロット変更割込み
19 ; タイムスロット値(変更値)
20 ; タイムスロット値(現在値)
22 ; テーブル情報(現在値)
23 ; テーブル情報(変更値)
25 ; 起動要求タスク
27 ; スケジューリングモード信号

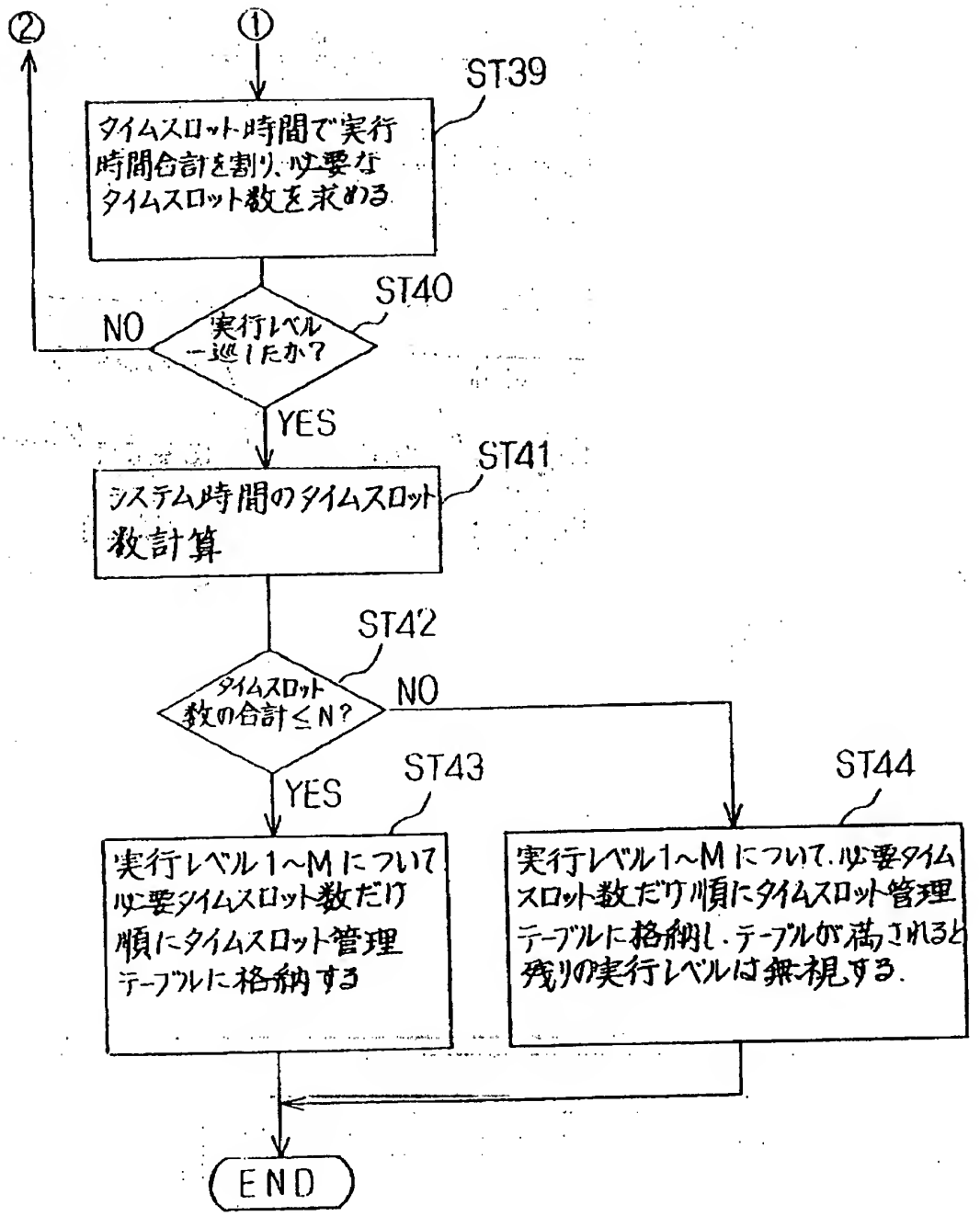
【図22】



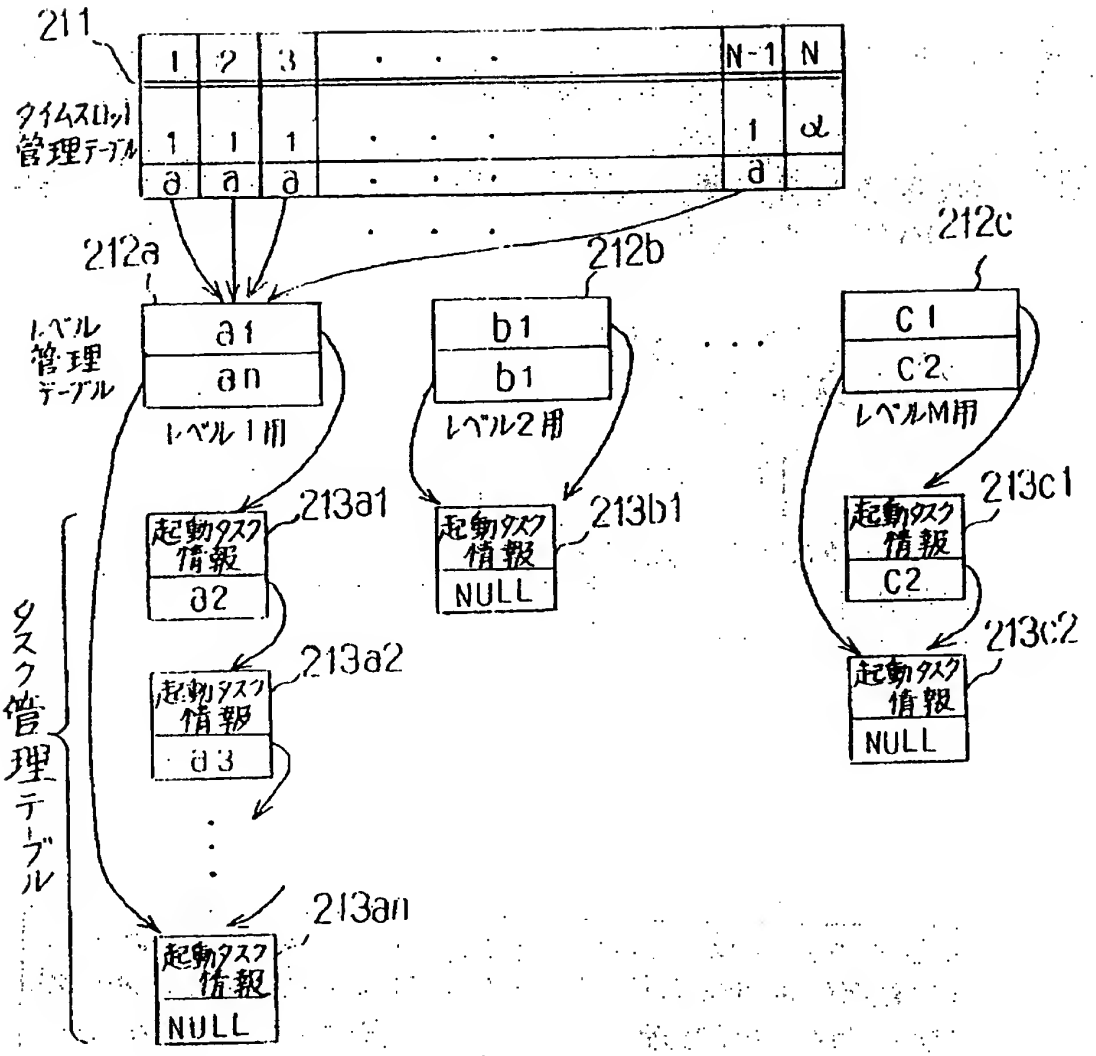
【図3】


START
ST31
全てのタスクを周期毎の実行レベルに分類
ST32
実行レベルiの周期＞最小基準周期?
NO
YES
ST33
スケジューリングモード←周期的スケジューリング
ST34
スケジューリングモード←プライオリティスケジューリング
ST35
スケジューリングモード＝周期?
NO
YES
ST36
全てのタイムスロットに実行レベル1を割当てる
END
ST37
実行レベル 1～Mについて
ST38
最小基準周期単位にタスクの実行時間を換算し、その合計を求める.
②
①
図14へ続く

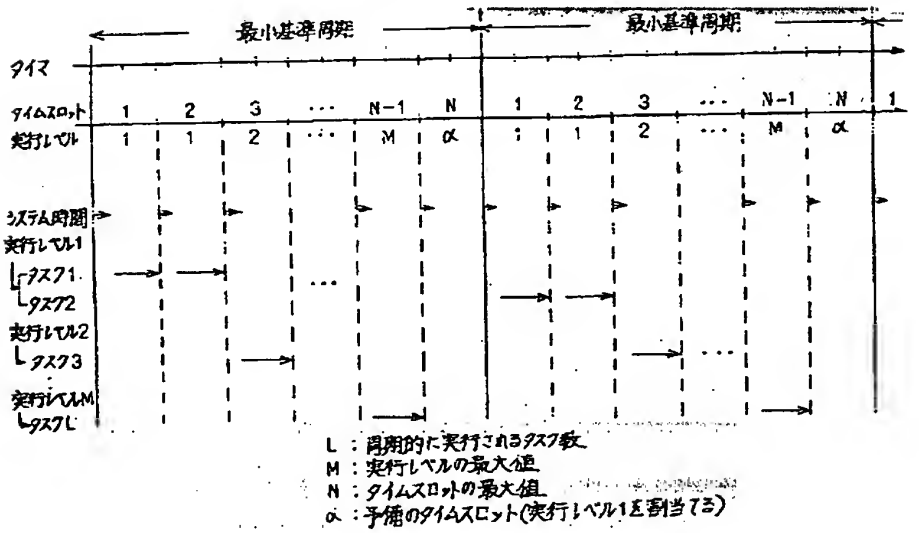
【図4】


②
①
ST39
タイムスロット時間で実行時間合計を割り、必要なタイムスロット数を求める
ST40
NO
実行レベル一巡したか?
YES
ST41
システム時間のタイムスロット数計算
ST42
タイムスロット数の合計≦N?
NO
YES
ST43
実行レベル1～Mについて必要タイムスロット数だけ順にタイムスロット管理テーブルに格納する
ST44
実行レベル1～Mについて、必要タイムスロット数だけ順にタイムスロット管理テーブルに格納し、テーブルが満されると残りの実行レベルは無視する。
END

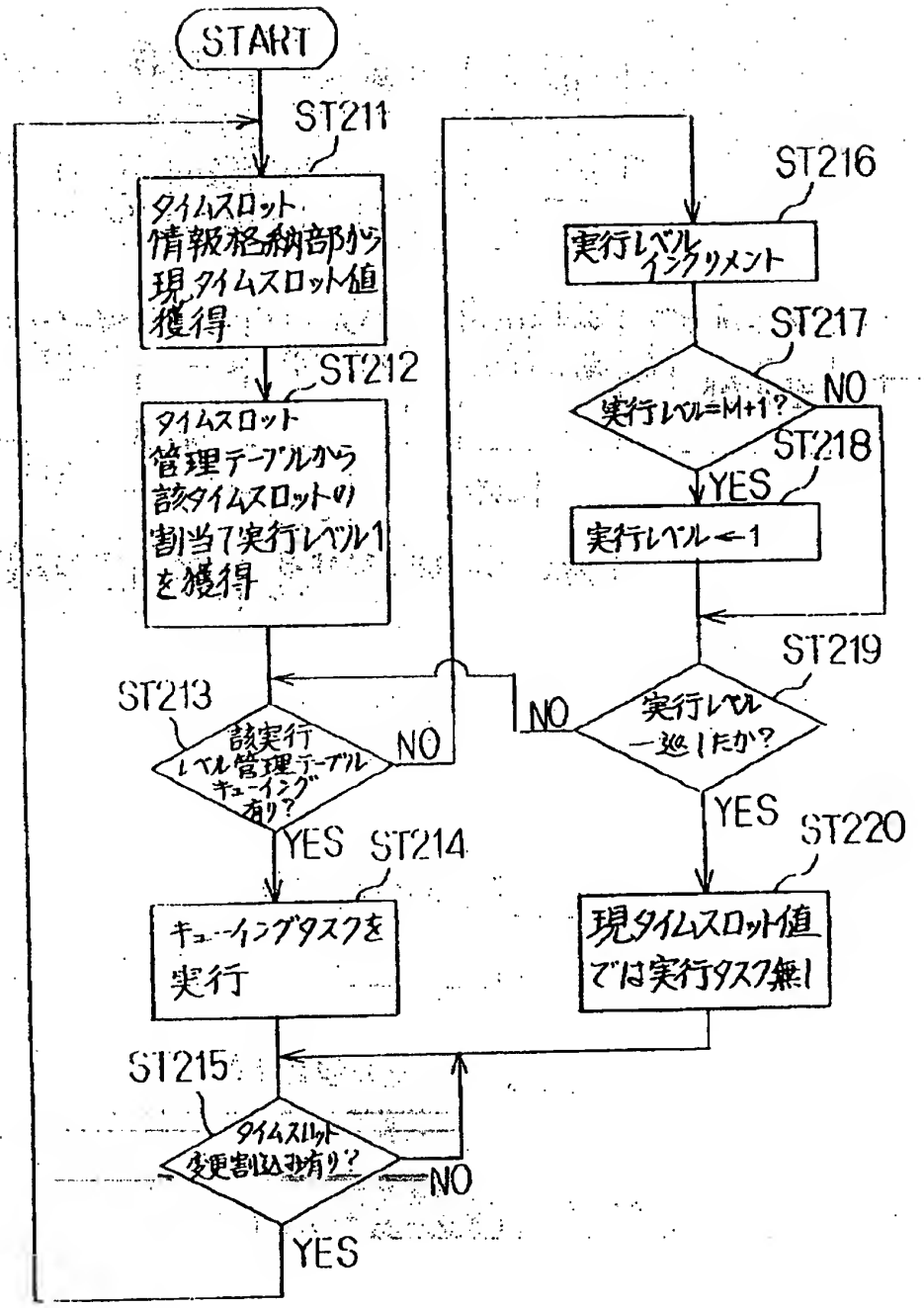
【図5】


211
タイムスロット管理テーブル
1
2
3
N-1
N
α
212a
212b
212c
レベル管理テーブル
a1
an
レベル1用
b1
b1
レベル2用
C1
C2
レベルM用
タスク管理テーブル
起動タスク情報
a2
a3
NULL
213a1
213a2
213an
213b1
213c1
213c2

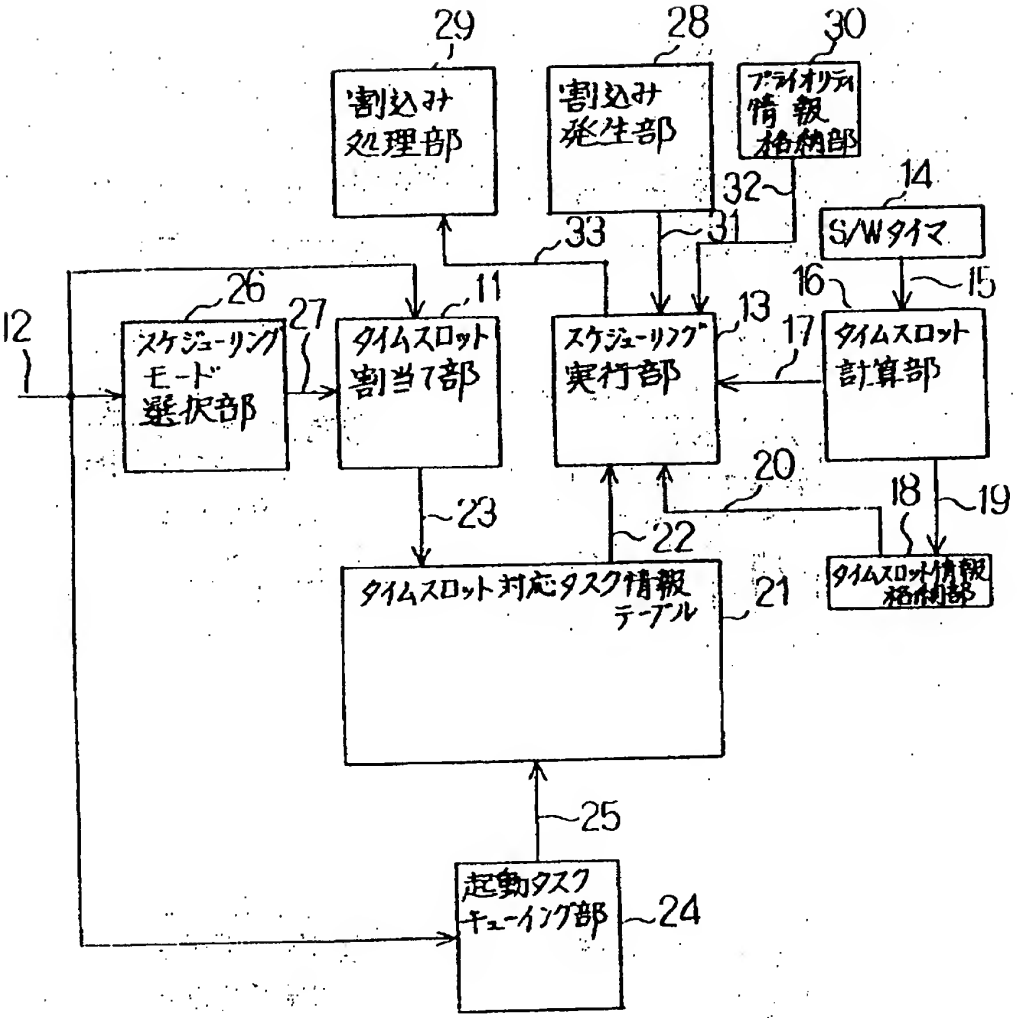
【図16】


最小基準周期
最小基準周期
タイマ
タイムスロット
実行レベル
システム時間
実行レベル1
タスク1
タスク2
実行レベル2
タスク3
実行レベルM
タスクL
L：周期的に実行されるタスク数
M：実行レベルの最大値
N：タイムスロットの最大値
α：予備のタイムスロット(実行レベル1に割当てる)

【図6】


START
ST211
タイムスロット情報格納部から現タイムスロット値獲得
ST212
タイムスロット管理テーブルから該タイムスロットの割当て実行レベルを獲得
ST213
該実行レベル管理テーブルキューイング有り?
NO
YES
ST214
キューイングタスクを実行
ST215
タイムスロット変更割込み有り?
NO
YES
ST216
実行レベルインクリメント
ST217
実行レベル=M+1?
NO
YES
ST218
実行レベル←1
ST219
実行レベル一巡したか?
NO
YES
ST220
現タイムスロット値では実行タスク無し

【図8】

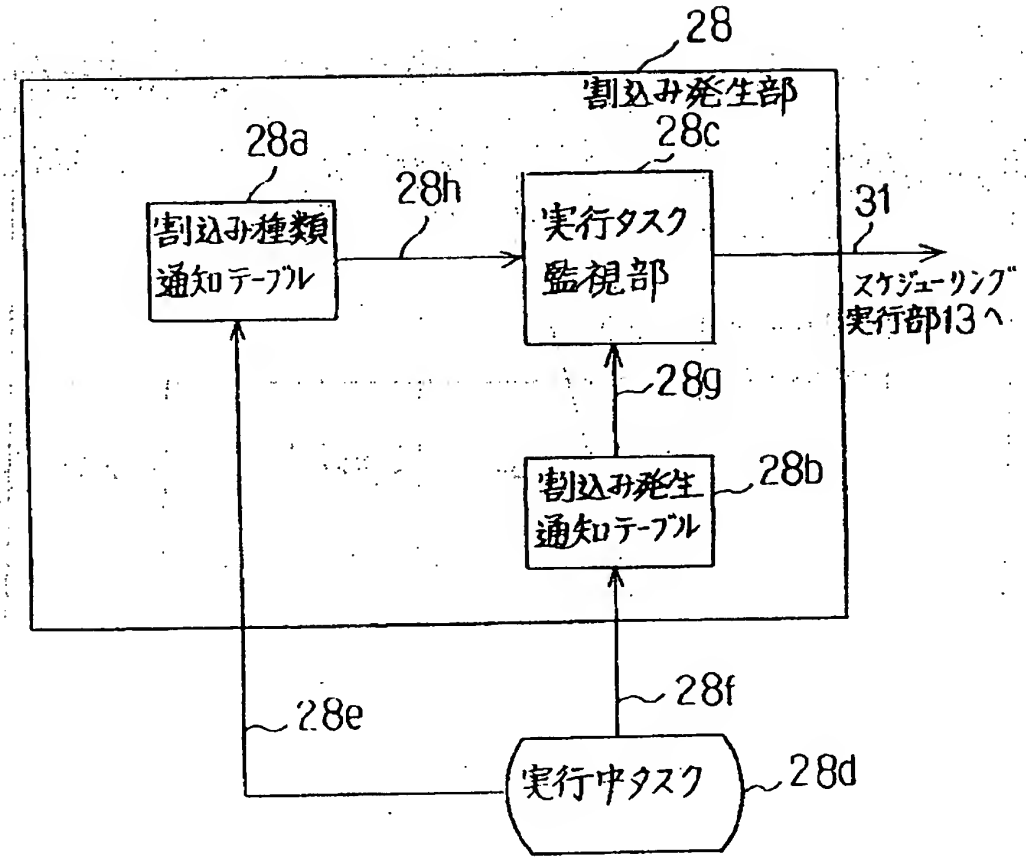


31；割込み（割込み種類を含む）

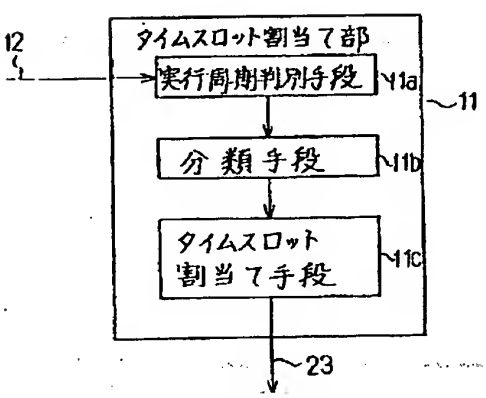
32；プライオリティ情報

33；割込み処理要求
（割込み種類を含む）

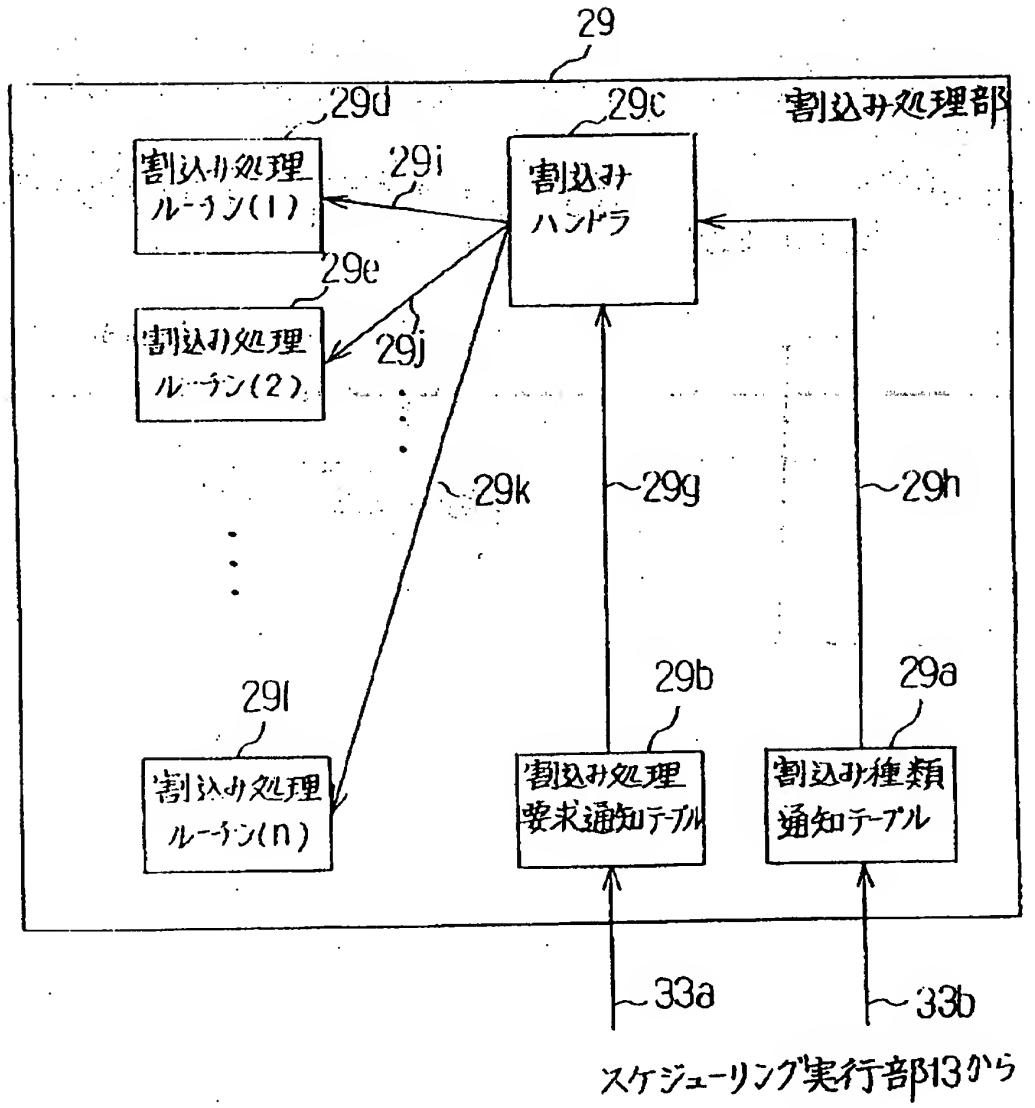
【図９】


28
割込み発生部
28a
割込み種類
通知テーブル
28h
28c
実行タスク
監視部
31
スケジューリング
実行部13へ
28g
割込み発生
通知テーブル
28b
28e
28f
実行中タスク
28d

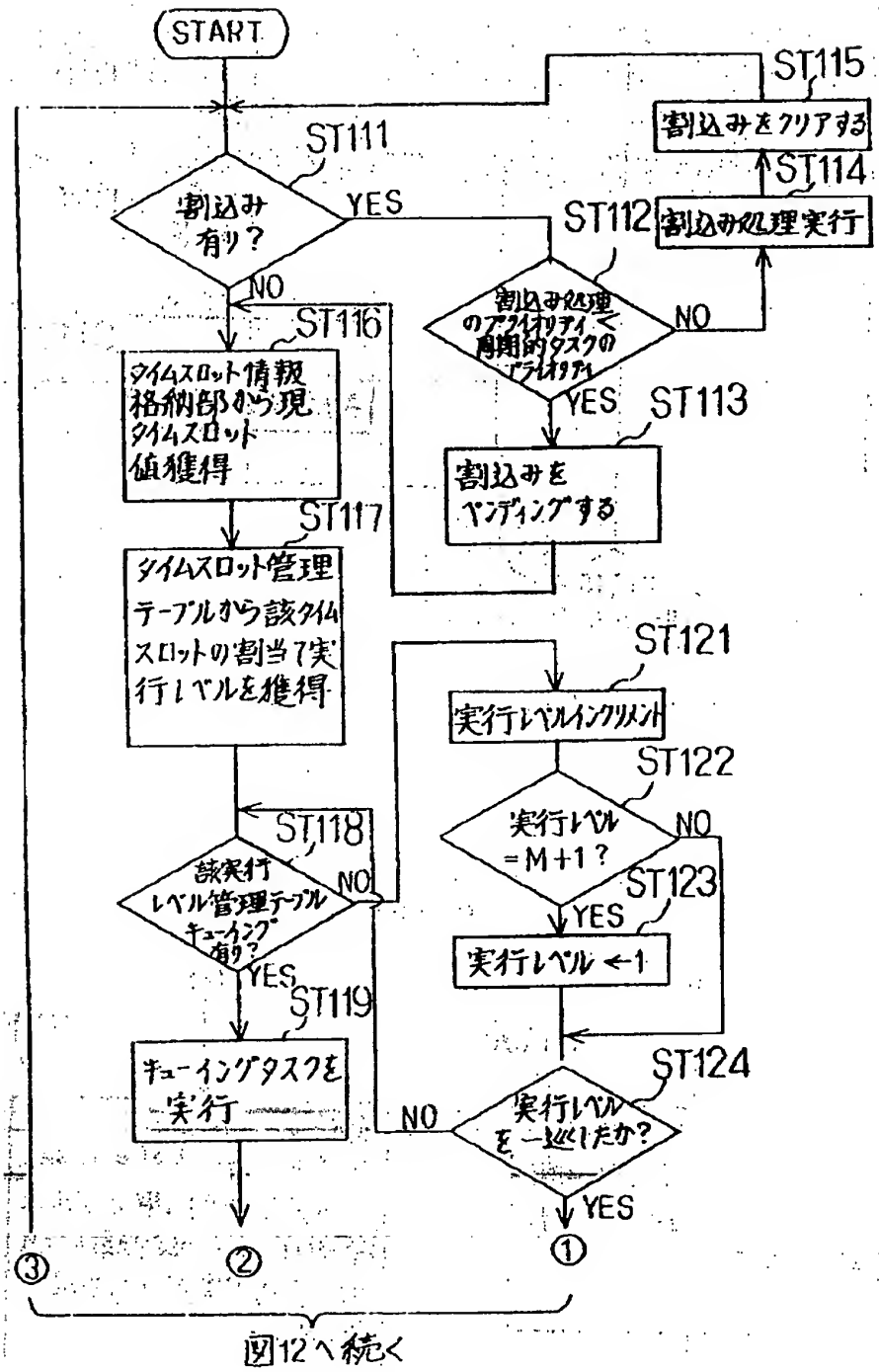
【図１７】


12
タイムスロット割当て部
実行周期判別手段
11a
11
分類手段
11b
タイムスロット
割当て手段
11c
23

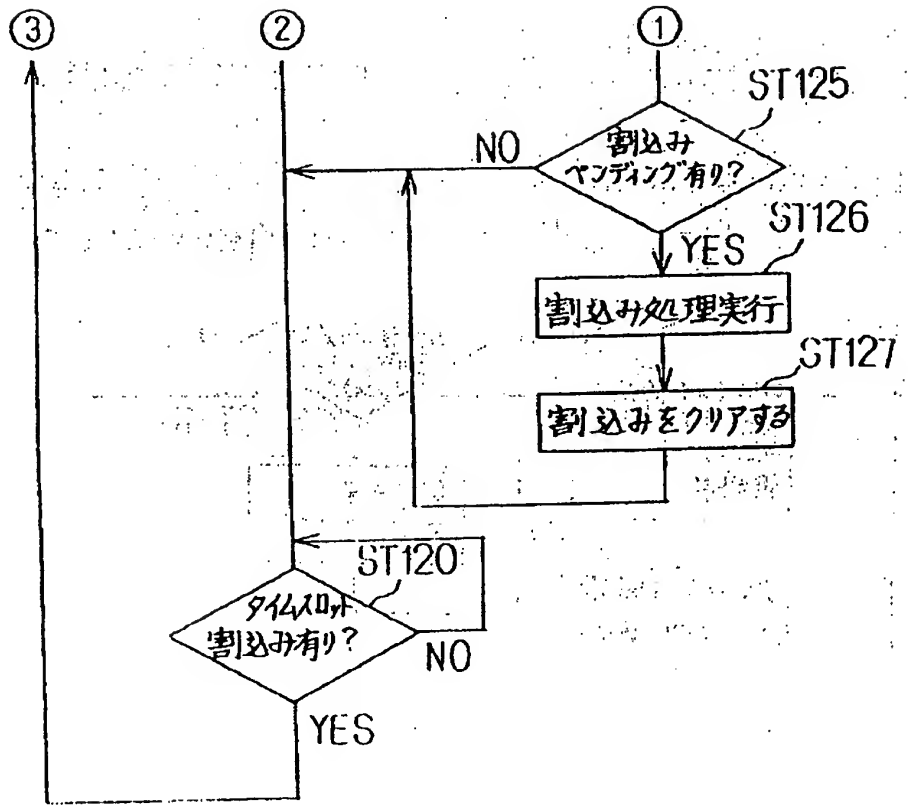
【図１０】


29
29d
29c
割込み処理部
割込み処理ルーチン(1)
29i
割込みハンドラ
29e
割込み処理ルーチン(2)
29j
29k
29g
29h
29l
29b
29a
割込み処理ルーチン(n)
割込み処理要求通知テーブル
割込み種類通知テーブル
33a
33b
スケジューリング実行部13から

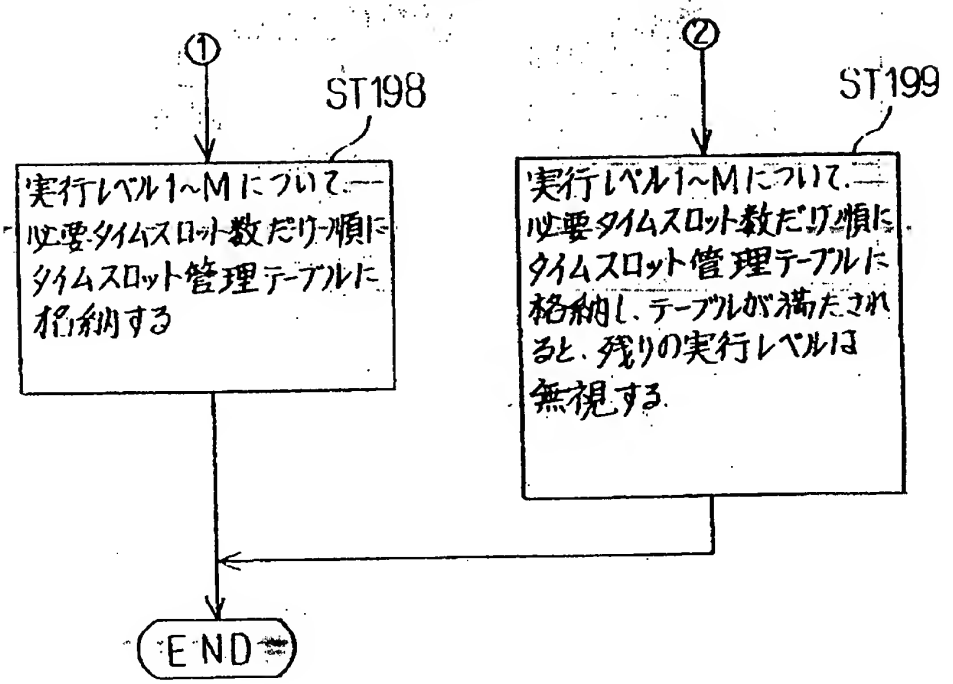
【図１１】


START
ST111
割込み有り？
YES
NO
ST112
割込み処理のプライオリティ＜周期的タスクのプライオリティ
NO
YES
ST113
割込みをペンディングする
ST114
割込み処理実行
ST115
割込みをクリアする
ST116
タイムスロット情報格納部から現タイムスロット値獲得
ST117
タイムスロット管理テーブルから該タイムスロットの割当て実行レベルを獲得
ST118
該実行レベル管理テーブルキューイング有り？
NO
YES
ST119
キューイングタスクを実行
ST121
実行レベルインクリメント
ST122
実行レベル＝M＋1？
NO
YES
ST123
実行レベル←1
ST124
実行レベルを一巡したか？
NO
YES
①
②
③
図12へ続く

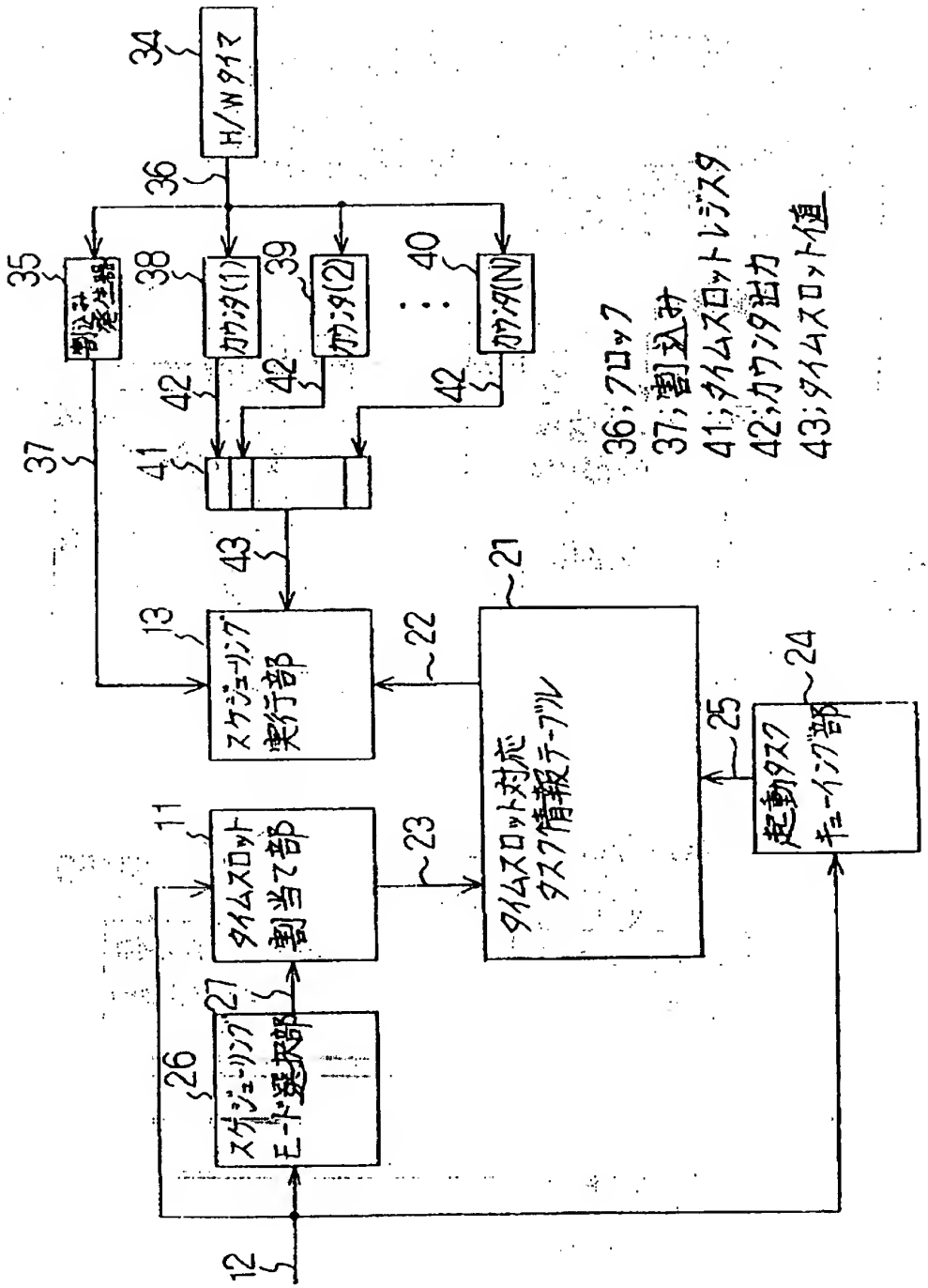
【図１２】


③
②
①
ST125
割込み
ペンディング有り?
NO
YES
ST126
割込み処理実行
ST127
割込みをクリアする
ST120
タイムスロット
割込み有り?
NO
YES

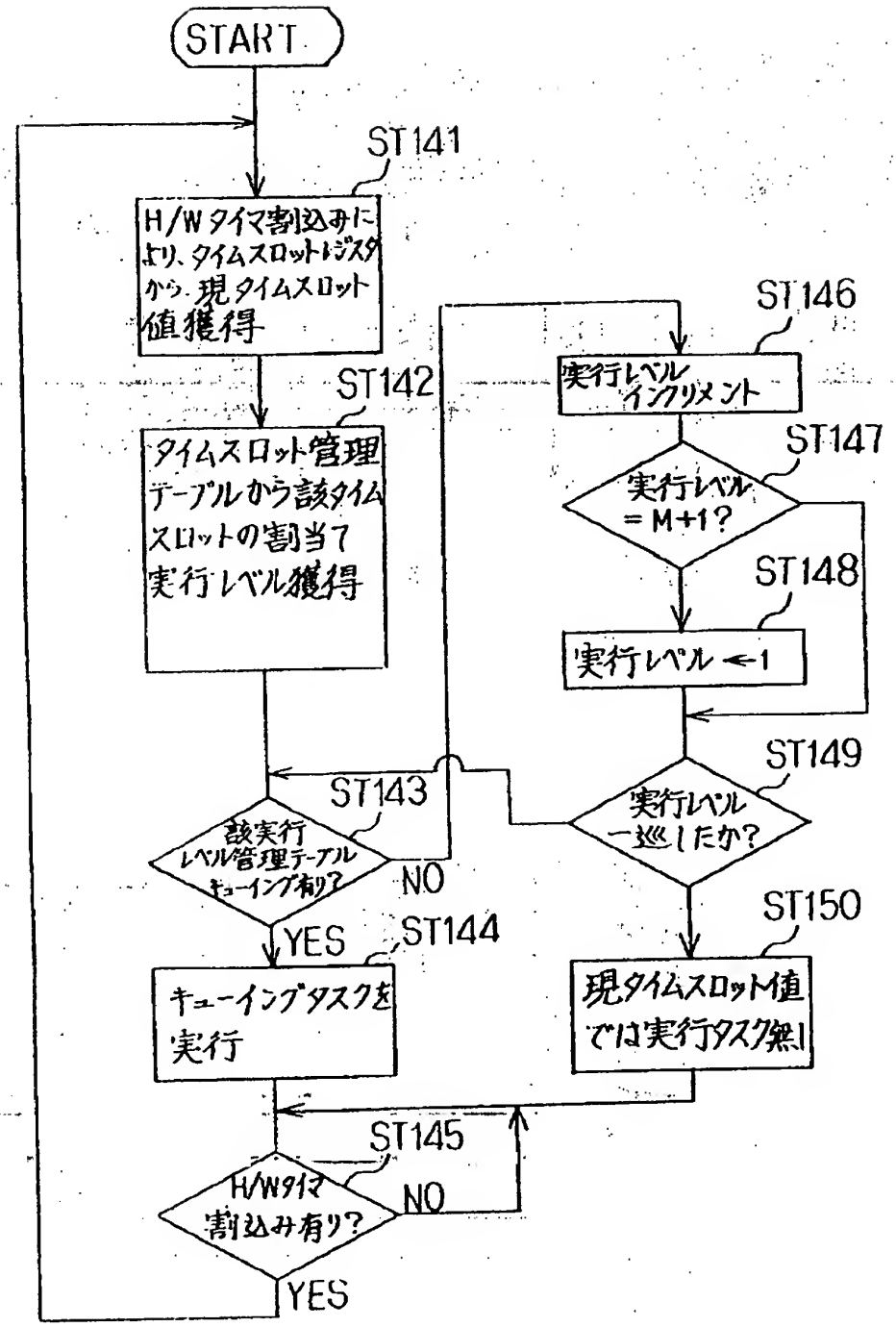
【図２０】


①
ST198
実行レベル1~Mについて、
必要タイムスロット数だけ順に
タイムスロット管理テーブルに
格納する
②
ST199
実行レベル1~Mについて、
必要タイムスロット数だけ順に
タイムスロット管理テーブルに
格納し、テーブルが満たされ
ると、残りの実行レベルは
無視する
END

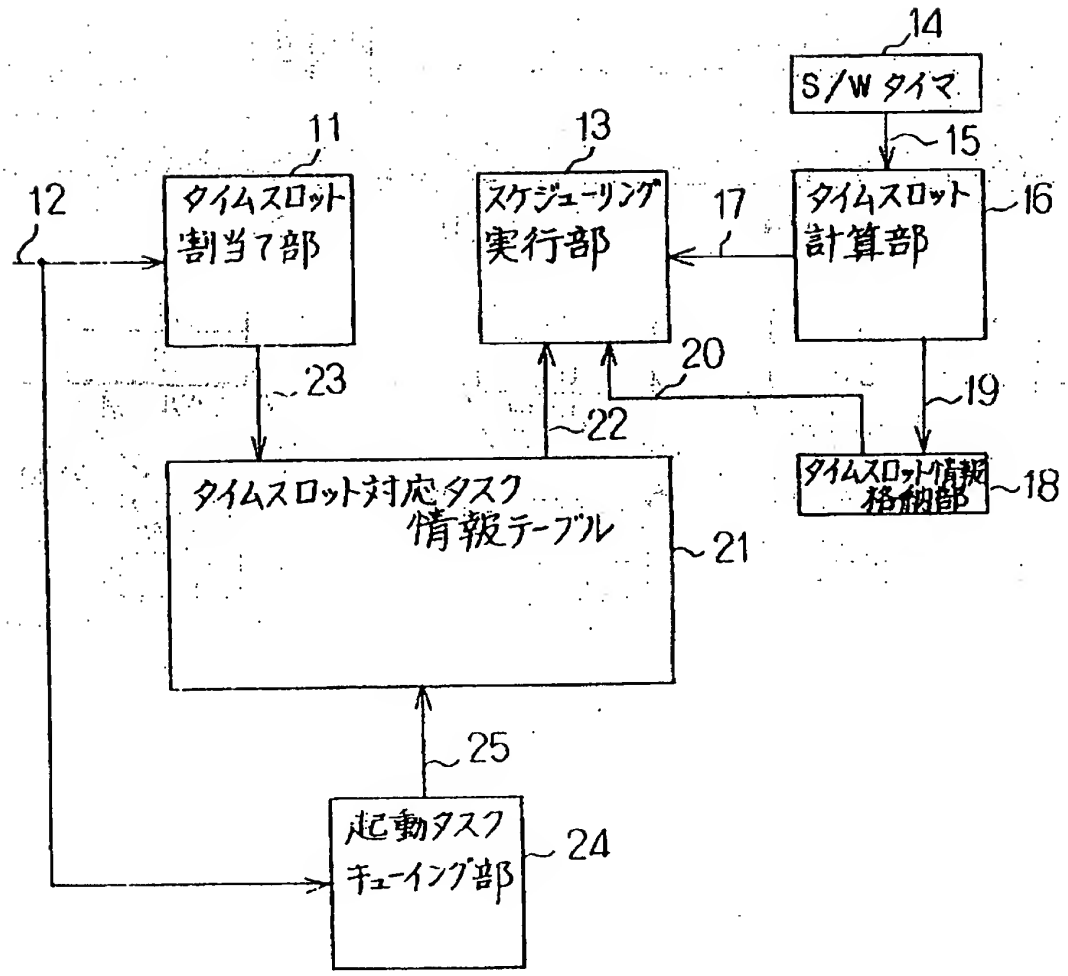
【図１３】


34
H/Wタイマ
36
35
割込み発生部
38
カウンタ(1)
39
カウンタ(2)
40
カウンタ(N)
42
37
41
43
13
スケジューリング実行部
22
21
タイムスロット対応タスク情報テーブル
25
24
起動タスクキューイング部
11
タイムスロット割当て部
23
26
スケジューリングエントリ選択部
27
12
36; ブロック
37; 割込み
41; タイムスロットレジスタ
42; カウンタ出力
43; タイムスロット値

【図１４】

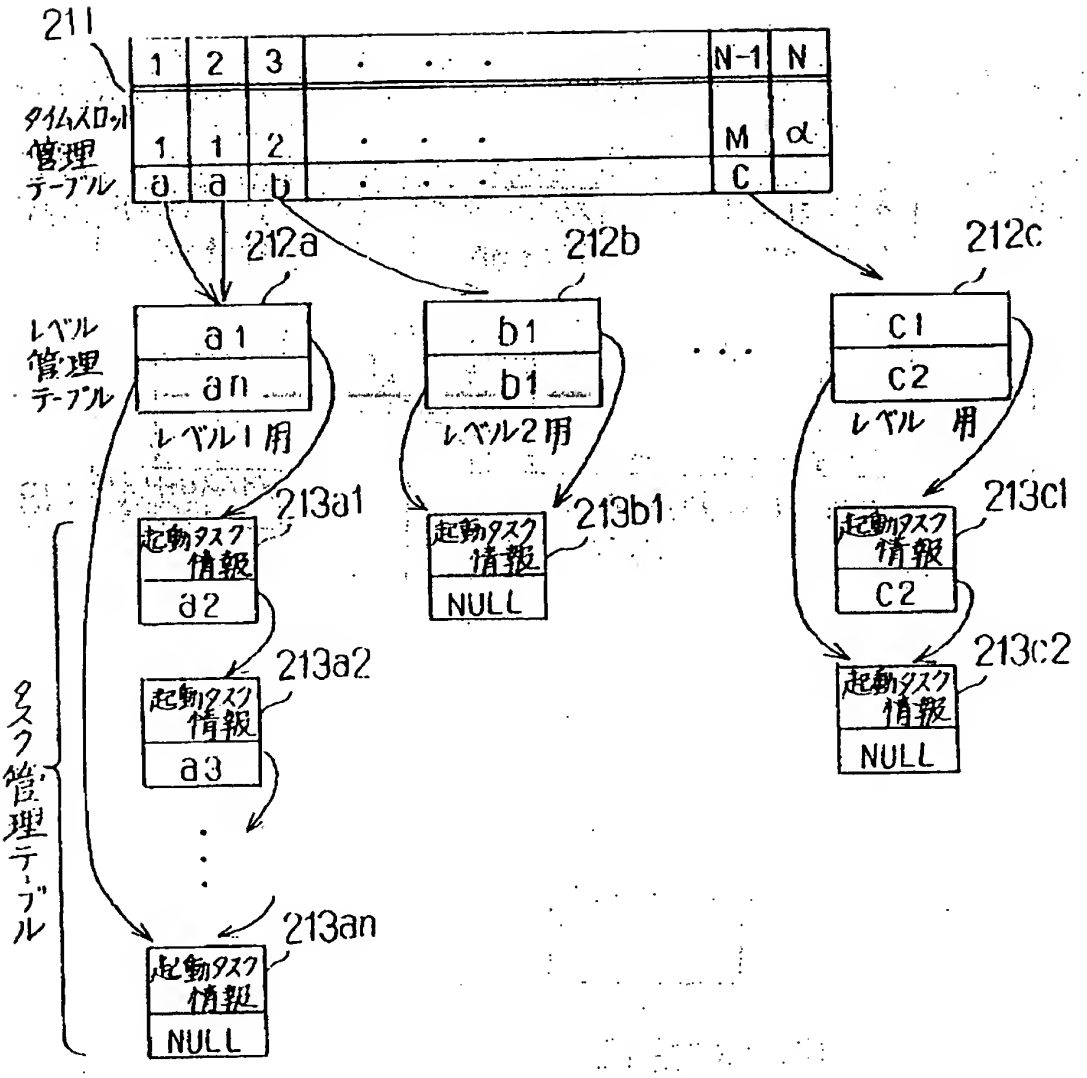

START
ST141
H/Wタイマ割込みにより、タイムスロットレジスタから現タイムスロット値獲得
ST142
タイムスロット管理テーブルから該タイムスロットの割当て実行レベル獲得
ST143
該実行レベル管理テーブルキューイング有り?
NO
YES
ST144
キューイングタスクを実行
ST145
H/Wタイマ割込み有り?
NO
YES
ST146
実行レベルインクリメント
ST147
実行レベル=M+1?
ST148
実行レベル←1
ST149
実行レベル一巡したか?
ST150
現タイムスロット値では実行タスク無し

【図15】

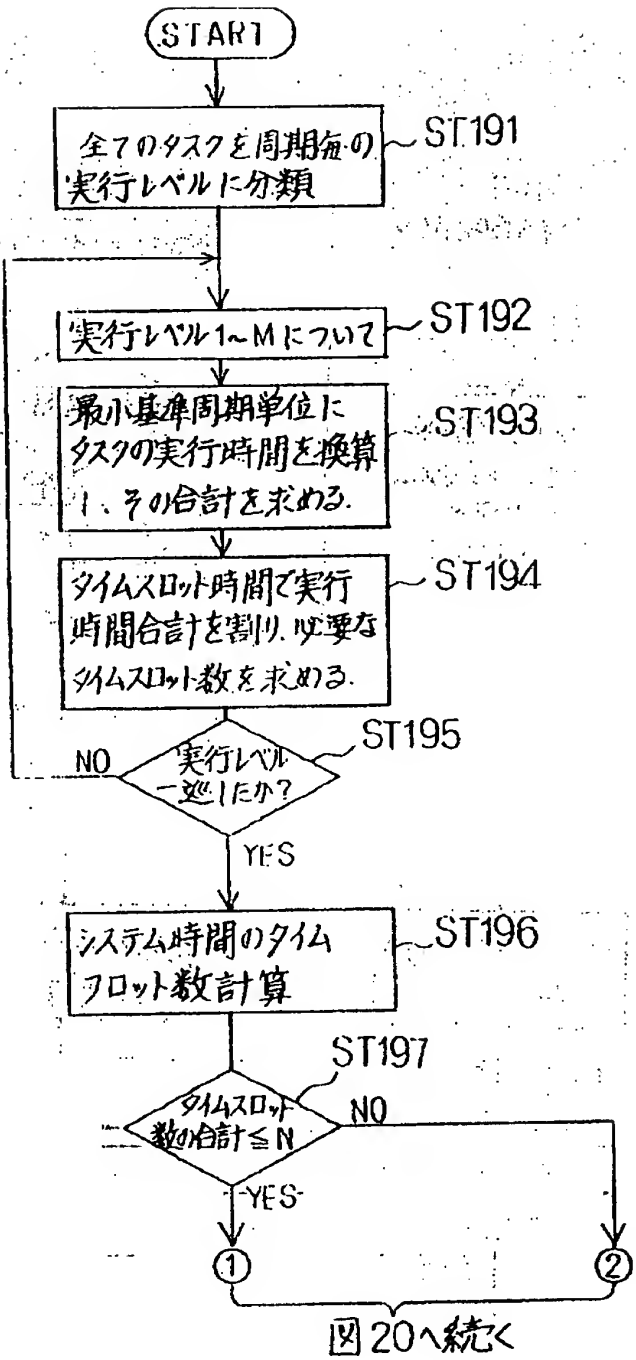


12; タスク情報
15; 定周期割込み
17; タイムスロット変更割込み
19; タイムスロット値（変更値）
20; タイムスロット値（現在値）
22; テーブル情報（現在値）
23; テーブル情報（変更値）
25; 起動要求タスク

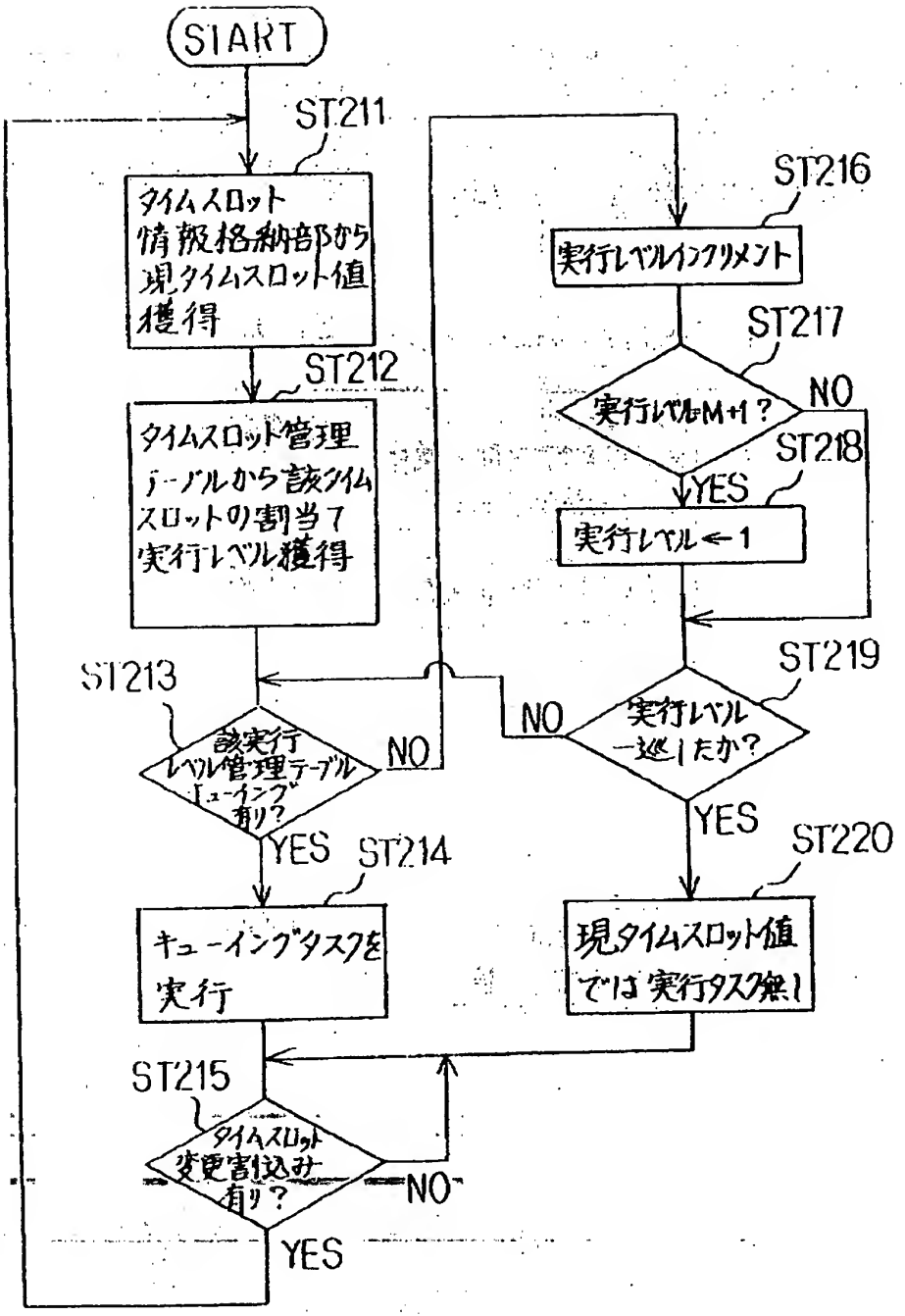
【図18】


211
1
2
3
N-1
N
タイムスロット管理テーブル
1
1
2
M
α
a
a
b
C
212a
212b
212c
レベル管理テーブル
a1
an
レベル1用
b1
b1
レベル2用
C1
C2
レベル用
213a1
213b1
213c1
起動タスク情報
a2
NULL
C2
213a2
213c2
a3
213an
タスク管理テーブル

【図19】


START
全てのタスクを周期毎の実行レベルに分類
ST191
実行レベル1～Mについて
ST192
最小基準周期単位にタスクの実行時間を換算し、その合計を求める.
ST193
タイムスロット時間で実行時間合計を割り、必要なタイムスロット数を求める.
ST194
実行レベル一巡したか?
ST195
NO
YES
システム時間のタイムスロット数計算
ST196
タイムスロット数の合計≦N
ST197
NO
YES
①
②
図20へ続く

【図２１】



【手続補正書】
【提出日】平成５年５月２０日
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】０００３
【補正方法】変更
【補正内容】
【０００３】まず、周期的スケジューリングの概念について、図１６を用いて説明する。図中の最小基準周期とは、周期的にタスクの起動をチェックする際の基準周期となるもので、周期的に実行されるアプリケーションタスク１～Ｌ（Ｌはタスク数）をスケジューリングする最小周期である。タイムスロットとは、上記最小基準周期を所定値Ｎ（ユーザがシステムに合わせて決定する）で割った時間単位で、タイムスロット時間毎に、スケジュ

ーラが実行タスク１～Ｌの選択を行う。実行レベルとは、実行タスク１～Ｌを周期毎に分類した複数のタスクの集合で、タイムスロット１～Ｎにその実行レベル１～Ｍが割当てられる（割当て方法は後述する）。スケジューラはスケジューリング実行時に、あるタイムスロットでそのタイムスロットに割当てられた該実行レベルの起動待ちタスクを優先的に実行する（実行レベルの選択方法は後述する）。

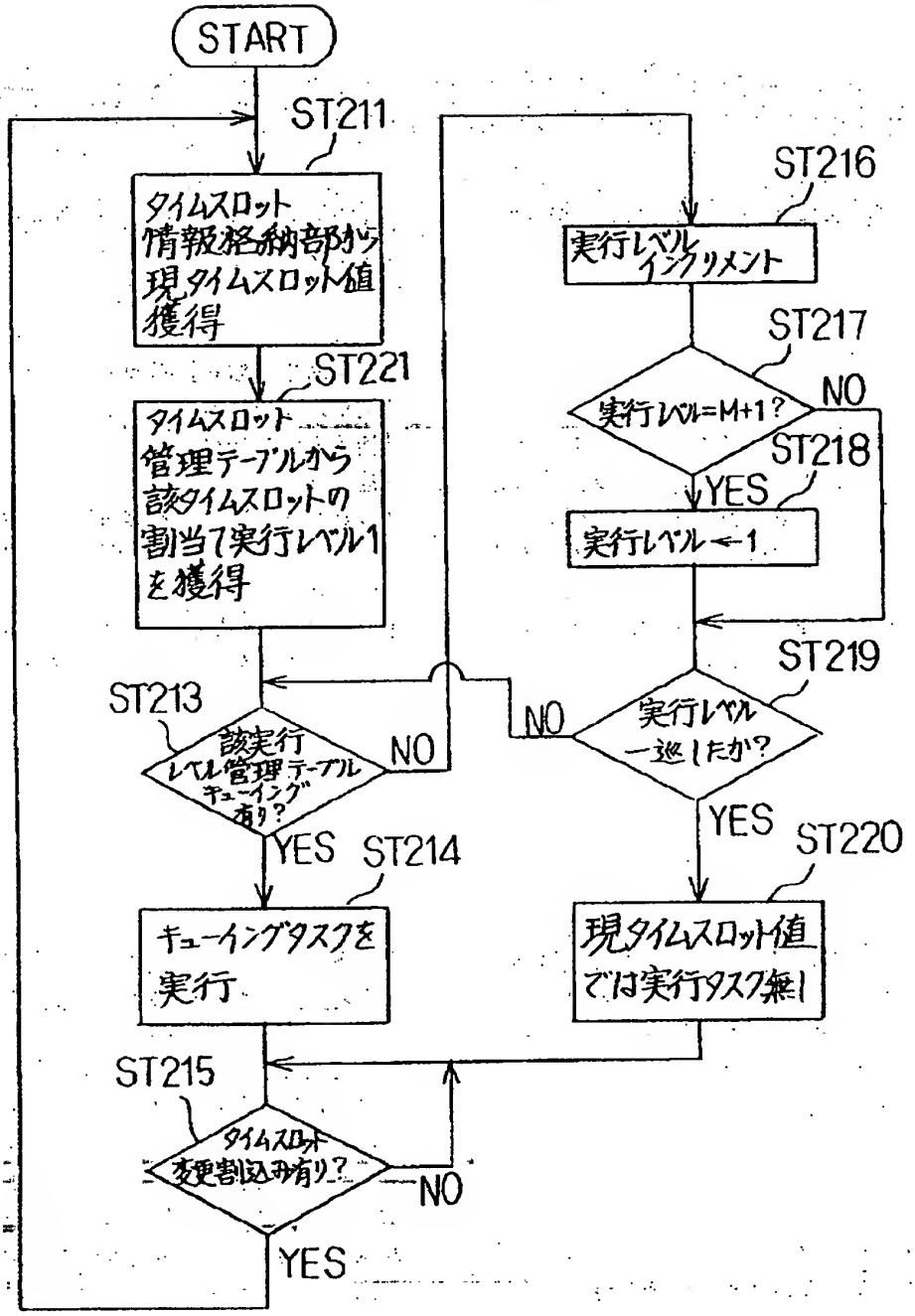
【手続補正２】
【補正対象書類名】明細書
【補正対象項目名】０００４
【補正方法】変更
【補正内容】
【０００４】図１６の例では、タイムスロット１とタイムスロット２に実行レベル１を、タイムスロット３に実行レベル２を、タイムスロット（Ｎ－１）に実行レベルＭを割当て、タイムスロットＮは、実行レベルの選択時間やタイムスロットの計算時間として、各タイムスロット処理に要するシステム時間のための予備のタイムスロット（α）であり、実行レベル１を割当てる。実行レベル１にはタスク１とタスク２の２つのタスクが、実行レベル２にはタスク３が、実行レベルＭにはタスクＬが所属している。スケジューリング時、タイマの値から、後述するタイムスロット計算部がタイムスロット値を計算する。最小基準周期毎に、実行周期が巡ってきた実行レベルのタスクが後述する起動タスクキューイング部により起動要求されるが、異なった実行レベルが同時に起動要求された場合、以下の様にタイムスロット毎に起動要求実行レベルの切換が実行される。タイムスロット１においては、実行レベル１が選択され、実行レベル１のタスク１が実行されている。タイムスロット２においても、実行レベル１が選択され、同様にタスク１が実行されている。タイムスロット３においては実行レベル２が選択され、実行レベル２の所属タスク３が実行される。タイムスロットＮは、予備のタイムスロットなので、最小基準周期内で未実行のタスクが実行される。以上の様な実行レベル選択が、最小基準周期毎に繰り返される。

【手続補正３】
【補正対象書類名】明細書
【補正対象項目名】００２０
【補正方法】変更
【補正内容】
【００２０】タイムスロット割当て後のスケジューリング実行は、従来の手段と同様の動作を行う。この時、プライオリティスケジューリングモードで図５に示したようにタイムスロット管理テーブル２１１が生成された場合は、従来例で示した図２１のフロー図は図６に示すようになり、全てのタイムスロットで実行レベル１を得て（ＳＴ２２１）、実行レベル１から図２２に従ってレベル管理テーブル２１２ａ～２１２ｃをチェックし、最初に発見したキューイングタスクを実行する。この結果、実行レベル１に最大のプライオリティを与え、実行レベルＭに最小のプライオリティを与えることになる。即ち、どのタイムスロットにおいても実行レベル１を獲得し、図２２の実行レベル探索に従い実行タスクを決定するので、実行レベルの小さいタスクが終了しない限り、大きいタスクは実行されない。この結果、実行レベルの小さいタスクほど高いプライオリティを与えるプライオリティスケジューリングが実現される。この場合、従来例で示した図１６は図７に示すようになる。

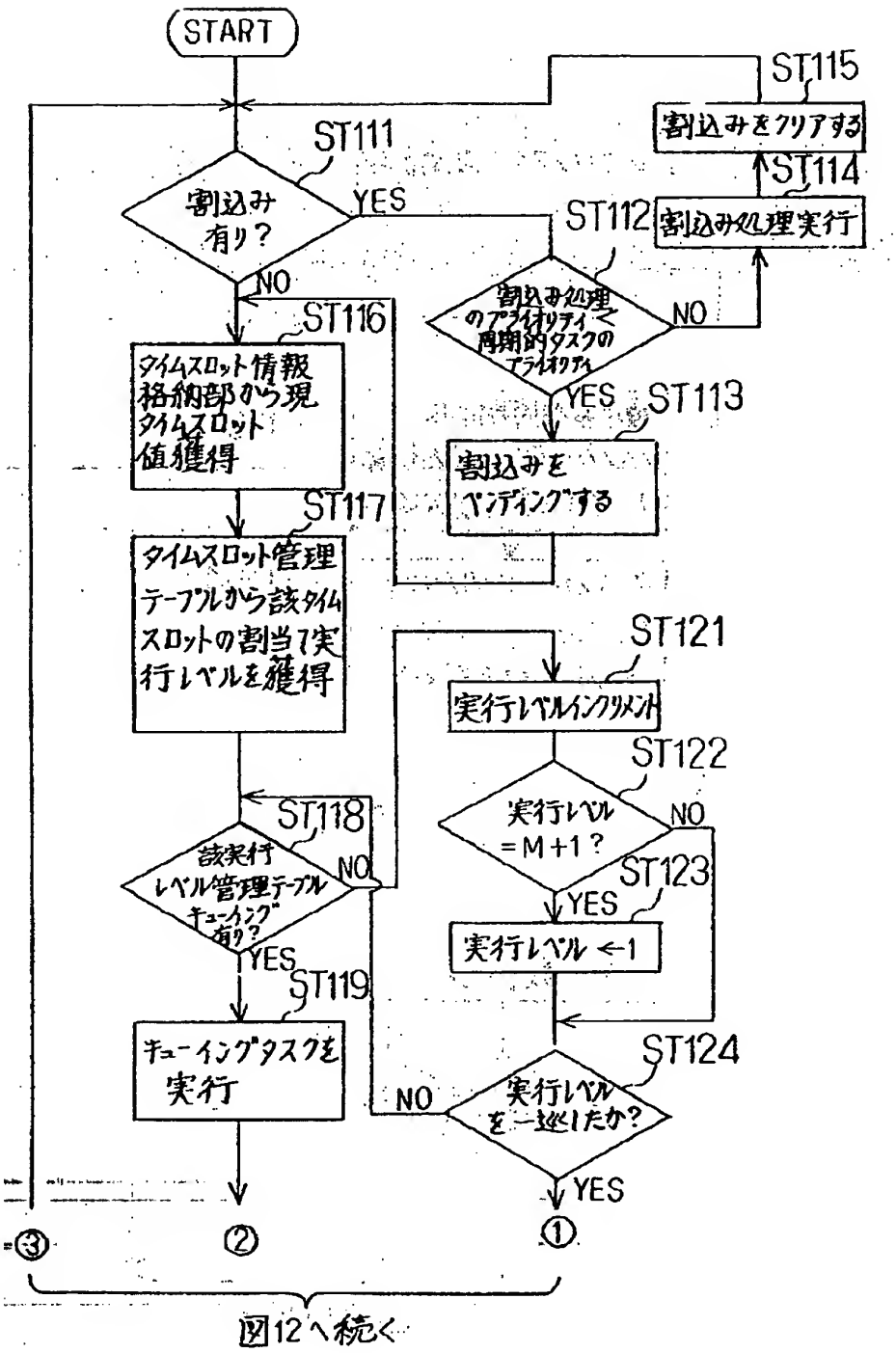
【手続補正４】
【補正対象書類名】明細書
【補正対象項目名】００２５
【補正方法】変更
【補正内容】
【００２５】次に、本実施例のスケジューリング動作について、図１４のフロー図を用いて説明する。Ｈ／Ｗタイマ３４のクロック３６を用い、割込み発生器３５はタイムスロット時間毎にスケジューリング実行部１３に割込み３７を発生する。カウンタ（１）３８からカウンタ（Ｎ）４０はＨ／Ｗタイマ３４のクロック３６を利用して、それぞれ定められたタイムスロット値１～Ｎをカウントし、最小基準周期毎にタイムスロットレジスタ４１の対応ビットをセットする。スケジューリング実行部１３は、割込み発生器３５からタイムスロット時間毎の割込み３７が入るとタイムスロットレジスタ４１を見に行き、セットされたビットからタイムスロット値４３を得る（ＳＴ１４１）。スケジューリング実行部１３は、タイムスロット値を獲得後、タイムスロットレジスタ４１をリセットする。このタイムスロット値を用いて、実施例１と同様のスケジューリング処理を行ない、Ｈ／Ｗタイマ割込みを待って（ＳＴ１４５）、次のタイムスロット値の獲得へ戻り、以上の動作を繰り返す。タイマにＨ／Ｗタイマを用い、レジスタに自動的にタイムスロット値をセットすることによって、システムにかかる負荷を軽減し、処理速度の向上を実現できる。即ち、実施例１では、オペレーティングシステムのＳ／Ｗタイマ機能を用い、現在の時刻を獲得する処理，タイムスロットの計算，スケジューリング実行部１３へのタイムスロット変更割込みの通知といった処理を全てソフトウエアで実現している。この場合、タイムスロットの前回値の保持が必要な上、オーバヘッド時間（図７のシステム時間）が大きくなるという欠点がある。これに対し、本実施例のように、以上の機能を、Ｈ／Ｗタイマを用いて割込み発生器，カウンタ及びレジスタなどのハードウェアで実現した場合、タイムスロットの前回値保持が不要となる上、オーバヘッドを低減でき、高速なスケジューリングが可能となる。

【手続補正５】
【補正対象書類名】図面

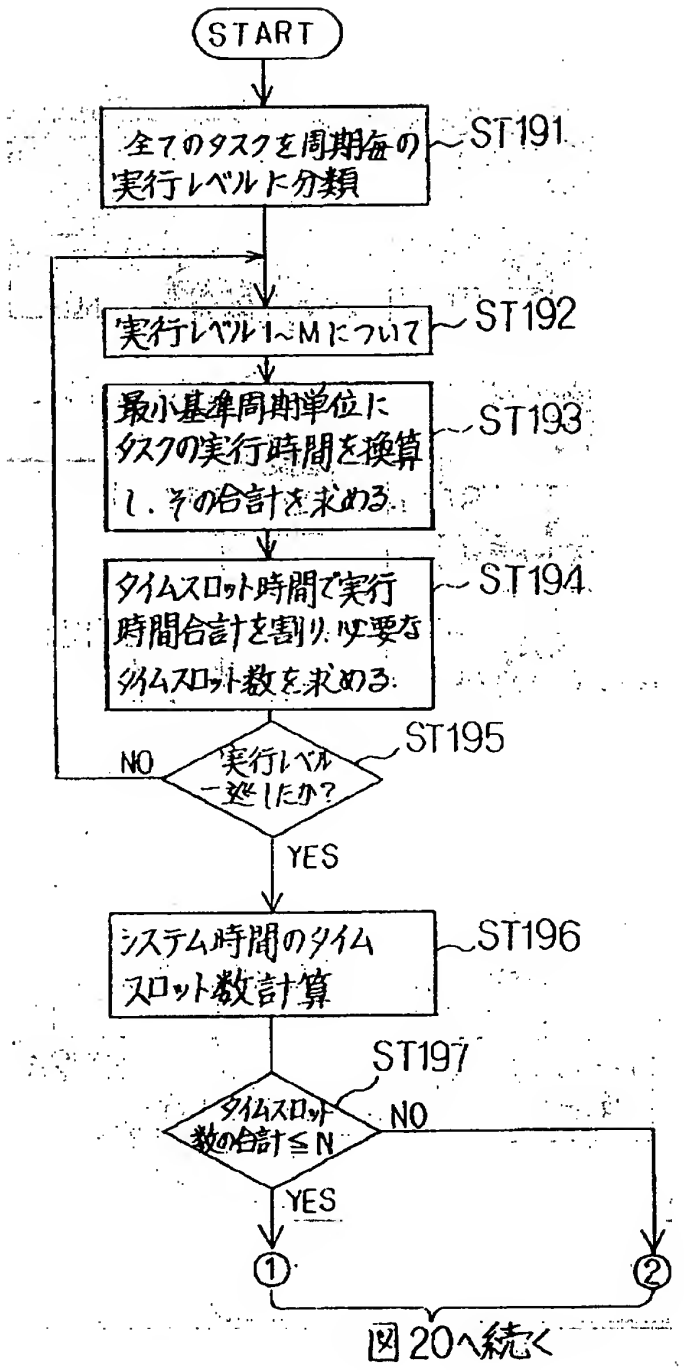
【補正対象項目名】図６
【補正方法】変更
＊【補正内容】
＊【図６】



【手続補正６】
【補正対象書類名】図面
【補正対象項目名】図１１
【補正方法】変更
【補正内容】
【図１１】



【手続補正７】
【補正対象書類名】図面
【補正対象項目名】図１９
【補正方法】変更
【補正内容】
【図１９】


START
全てのタスクを周期毎の実行レベルに分類
ST191
実行レベル1～Mについて
ST192
最小基準周期単位にタスクの実行時間を換算し、その合計を求める
ST193
タイムスロット時間で実行時間合計を割り必要なタイムスロット数を求める
ST194
実行レベル一巡したか?
ST195
NO
YES
システム時間のタイムスロット数計算
ST196
タイムスロット数の合計≦N
ST197
NO
YES
①
②
図20へ続く